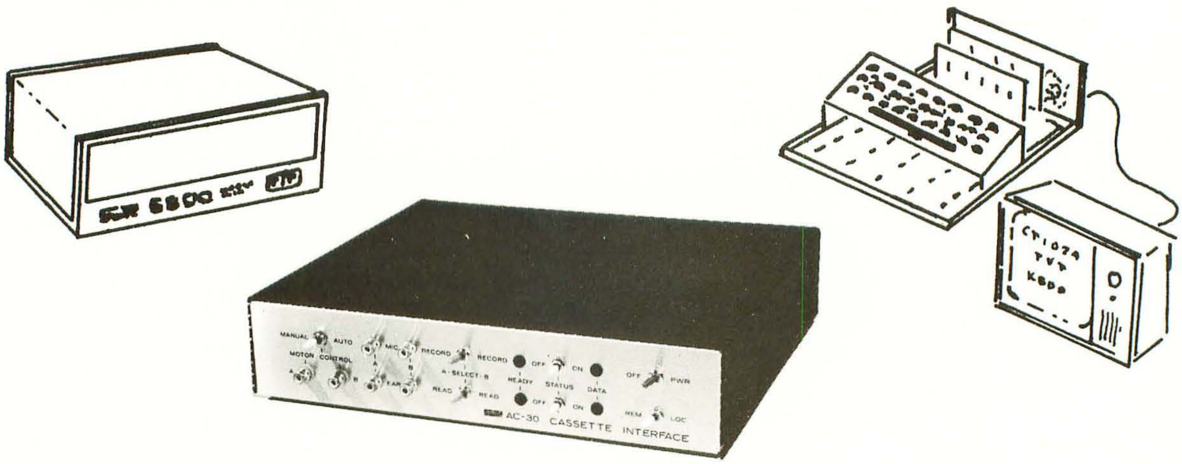
ISSUE NUMBER 11 PRINTED IN USA JULY 1976 $1.50

BYTE

the small systems journal



SUPPLIES THE MISSING LINK
AC-30 CASSETTE INTERFACE



Been looking for a practical way to input and dump programs to your computer? Well your search is over.

With our new AC-30 Cassette Interface you will be able to store and input program data to any computer system having RS-232 serial interfaces and a UART circuit having an accessible 16X clock frequency. Data format is the "Kansas City" standard which was selected for its tolerance of speed variations in the recording device. The AC-30 may be used with any cassette recorder of reasonable quality.

If <u>both</u> your computer and terminal have accessible 16X UART clocks and will operate at 300 baud—as do our 6800 computer and CT-1024 terminal system—the AC-30 may be used between the terminal's serial interface and the computers control interface. This eliminates the need for a separate interface to drive the cassette unit. It also allows you to use the computer system's tape load and dump routines built into Mikbug® or similar ROM software.

Independent control circuits are provided for <u>two</u> audio cassette recorders (not included in the kit). One recorder's tape may be read while the second is recording a new updated tape; making it possible to generate new program tapes, data tapes and to create program object tapes while reading and assembling program source tapes. The operating mode for each recorder is selected by switches on the front panel and LED indicators show the mode that is selected at any particular time. Computer controlled record, play and motor control commands may be used with this system if they are available from the terminal being used. This feature is available on our CT-1024 terminal if the CT-CA cursor control card is installed.

The AC-30 is housed in a 12¾" x 3" x 12½" aluminum chassis. It is powered by a self contained 115/230 Volt AC 50-60 Hz power supply. Data is FSK format using 1200 Hz and 2400 Hz at a 300 baud data rate. Recorder speed tolerance need be only ± 20%.

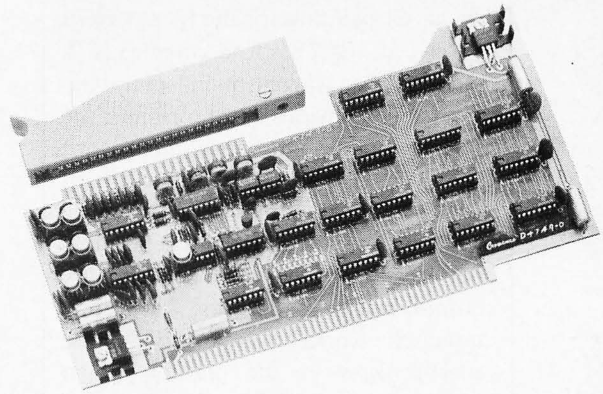
® Trademark Motorola

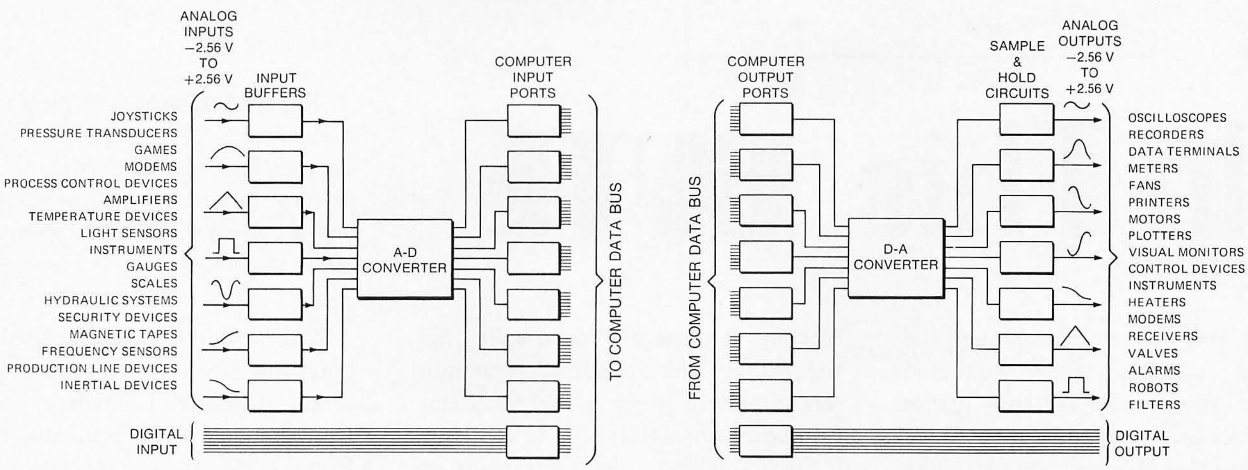
AC-30 Cassette Interface Kit $79.50 ppd

Southwest Technical Products Corporation

219 W. Rhapsody
San Antonio, Texas 78216

How to use your computer with analog signals





Now you have a way to get analog information into and out of your microcomputer. It's an easy, fast, and unbelievably inexpensive way.

It's Cromemco's new D+7A™ high-performance I/O module which gives you:

- 7 channels of 8-bit analog-to-digital conversion (to input analog data to the computer)
- 7 channels of digital-to-analog conversion (to output computer data in analog form)
- an 8-bit parallel I/O port to input and output data in digital form.
- a fast conversion time of 5 microseconds.

A MULTITUDE OF USES

The D+7A makes it easy to use your computer for the jobs you want it to do—such as process control, digital filtering, games, oscilloscope graphics, speech recognition, speech and music synthesis.

The D+7A lets you input and output analog data with all sorts of devices: joysticks, ham radio gear, measurement instruments, machine tools, transducers, control systems, motors, recorders, and plotters, to name just a few.

NO FURTHER SOFTWARE NEEDED

The D+7A I/O plugs directly into the Altair 8800 or IMSAI 8080 microcomputers. Analog signal range is from −2.56 to +2.56 volts (20-millivolt increments) on both input and output sides.

Simple "Input" and "Output" instructions initiate A/D conversion and read in or out the ensuing 8 bits of data. No further software is required. During conversion the D+7A holds down the computer "Ready" line.

Addresses of the input and output ports are jumper-wire selectable in blocks of 8. Sample-and-hold circuitry is used to "latch" the analog outputs.

LOW-PRICED

The low price of the D+7A is a result of Cromemco's design leadership. The D+7A and all Cromemco peripherals are of advanced computer-grade quality. The D+7A is solder-masked and printed with full legend for easy, error-free assembly.

AT COMPUTER STORES/MAIL

You can get the D+7A at computer stores in either kit or assembled form.

Or order directly by mail from Cromemco. Delivery is from stock to 30 days. The D+7A is certain to be popular so order now.

D+7A™ I/O kit $145
D+7A™ I/O assembled $245

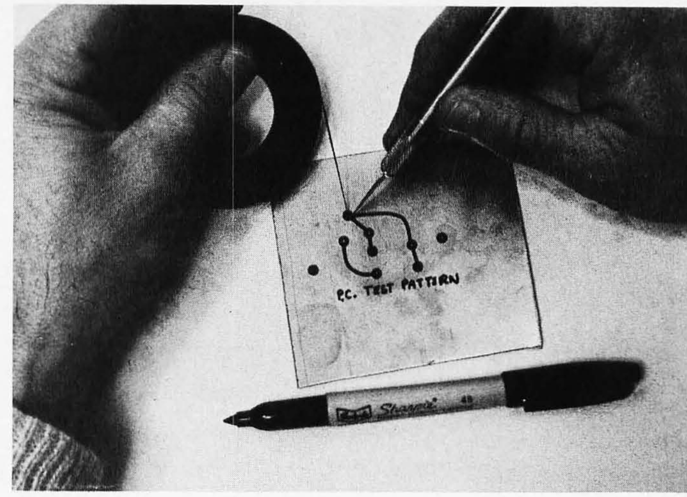
Each D+7A includes a connector to connect to the 8 input and 8 output ports.
Shipped prepaid if fully paid with order.
California users add 6% sales tax.
Mastercharge and BankAmericard accepted with signed order. Please include card number and expiration date.

Cromemco
Specialists in computer peripherals
One First St., Los Altos, CA 94022 • (415) 941-2967



page 6

We commissioned Robert Tinney to do an oil painting for the Bicentennial cover of BYTE. The theme is a humorous anachronism in several respects. The obvious anachronism is the 18th century philosopher and patriot sitting at a very 20th century hard copy terminal, holding a reference volume, America's first best seller. It is not clear how the quill pen and ink output mechanism of the hard copy terminal works. There are more anachronisms in the picture, which we'll leave to BYTE readers to point out. A full-size poster of the original painting in color is available from BYTE (see page 96).



page 58

In This BYTE

Robert Suding asks **"Why Wait?"** in a rhetorical fashion, and proceeds to demonstrate his schematic of a fast cassette interface which uses software and a one bit IO port to implement an audio cassette system.

Have you ever looked through the surplus catalogs and wondered whether those memory core planes and stacks advertised could be used for anything other than tea strainers? For theory and practical information on **Coincident Current Ferrite Core Memories** turn to James R Jones' article.

Bruce A Anderson describes his experiences **Assembling a Sphere** in his review of what rolled out of the production facilities in Bountiful UT last fall.

One of the most important questions people ask is "how do I learn about what a computer does?" One way to help out friends who are trying to get into the swing of things with programming is to implement a version of Charles Howerton's Educator-8080 program so that they can interactively **Explore an 8080 with Educator-8080.**

A thorough explanation of the instruction set should accompany any product intended for wide distribution. An example of such an explanation is provided by Nat Wadsworth's **Machine Language Programming for the "8008" and Similar Microcomputers,** a manual which is sold by Scelbi Computer Consulting Inc. In this issue is the first of three direct reprints from that manual: Chapter 1 which describes the 8008 instruction set.

One of the problems of interfacing unknown electronics is figuring out how to accomplish the match. Ken Barbier built a character generator, went out and bought a TV set, then faced the problem of building a driver for the TV. The result was **The "Ignorance is Bliss" Television Drive Circuit.**

While not really promising the entire big blue sky, when you **Put the "Do Everything" Chip in your Next Design** you'll end up with a computer that has five separate programmable real time clocks, standard serial communications data rates from 110 baud to 9600 baud, automatic generation of an 8080's RST n interrupt vectors, an 8 bit parallel output and an 8 bit parallel input port. Turn to Robert Baker's latest article to find out about this nifty chip.

What's it like to be isolated from bountiful US surplus markets? In a sense, it means a relative isolation from modern LSI products, as Dr Michael N Hayes reports on his experiences in Tokyo and Manila in December 1975. Read his report on **Surplus Electronics in Tokyo and Manila** in this issue.

There are many ways to wire a circuit. The most common manufacturing method is printed wiring. But you can also **Make Your Own Printed Circuits** at home, using techniques described by James Hogenson in his article.

One of the most interesting applications of computers is in the area of graphic outputs. Using a vector CRT or a plotter, drawing pictures of mathematically generated abstractions or simple cartoons can be the beginning of hours of fun. But **A Plot Is Incomplete Without Characters** so Richard J Lerseth concocted some software described in his article on the generation of an ASCII character set (or special characters) for a plotter or vector display device.

BYTE #11

JULY 1976

In the Queue

Foreground

6 COINCIDENT CURRENT FERRITE CORE MEMORIES
Recycling — Jones

22 EXPLORE AN 8080 WITH EDUCATOR-8080
Software — Howerton

38 THE "IGNORANCE IS BLISS" TV DRIVE CIRCUIT
Hardware — Barbier

46 WHY WAIT? BUILD A *FAST* CASSETTE INTERFACE
Hardware — Suding

Background

18 ASSEMBLING A SPHERE
Review — Anderson

30 MACHINE LANGUAGE PROGRAMMING FOR THE "8008"—Chapter 1
Software — Wadsworth

40 PUT THE "DO EVERYTHING" CHIP IN YOUR NEXT DESIGN
Hardware — Baker

54 SURPLUS ELECTRONICS IN TOKYO AND MANILA
Travelogue — Hayes

58 MAKE YOUR OWN PRINTED CIRCUITS
Hardware — Hogenson

64 A PLOT IS INCOMPLETE WITHOUT CHARACTERS
Software — Lerseth

Nucleus

2 In This BYTE
4 The Trend Toward Hassle Free Products
39 Tool Box
44, 79 BYTE's Bits
74 Good Grief!
76, 102 What's New?
81 Software Bug of the Month 2
83 What Happened at Trenton
86 Clubs, Newsletters
90 Letters
95 Ask BYTE
96 Reader's Service
98 Classified Ads
100 BYTE's Bugs
104 Book Reviews
112 BOMB

staff

PUBLISHERS
Virginia Peschke
Manfred Peschke
EDITOR
Carl T Helmers, Jr
GENERAL MANAGER
Manfred Peschke
PRODUCTION MANAGER
Judith Havey
PRODUCTION ASSISTANT
Elizabeth Alpaugh
CIRCULATION
Deborah R Luhrs
DEALER CIRCULATION
Deena Zealy
PUBLISHERS ASSISTANTS
Cheryl Hurd
Carol Nyland
ADVERTISING
Elizabeth Alpaugh
Virginia Peschke
TYPOGRAPHY
Custom Marketing Resources, Inc
Goodway Graphics
Mary Lavoie
Taimi Woodward
PHOTOGRAPHY
Ed Crabtree
Custom Marketing Resources, Inc
ART
Mary Jane Frohlich
Bill Morello
PRINTING
Custom Marketing Resources, Inc
The George Banta Company
ASSOCIATES
Bob Baker
Dan Fylstra
Don Lancaster
Harold A Mauch
Chris Ryland
CONTEST EDITOR
Janice D Black

BYTE magazine is published monthly by BYTE Publications, Inc., 70 Main St, Peterborough, New Hampshire 03458. Subscription rates are $12 for one year worldwide. Two years, $22. Three years, $30. Second class postage paid at Peterborough New Hampshire 03458 and at additional mailing offices. Phone 603-924-7217. **Address editorial correspondence to Editor, BYTE, 70 Main St, Peterborough NH 03458.**

The Trend Toward Hassle Free Products

Editorial by Carl Helmers

One item which backyard entrepeneurs tend to overlook is the fact that the hardware and software engineering of computer systems which are both manufacturable and marketable is a complex process. I had a taste of the problems of working out the details of a system about the time BYTE became a much more compelling personal project in mid 1975. I quickly dropped any illusions of converting the designs I was working upon at that time into a manufactured product when BYTE with its own intricate and unique challenges came into being. But the period of time spent thinking about the problem of engineering a salable combination of hardware and software leaves an impression.

There are a myriad of details which must converge into a well defined, usable product for personal computing applications. These details are handled naturally and implicitly when the advanced amateur or full time engineer assembles a "one of a kind" system as a personal vehicle or engineering test bed. But documenting the critical parameters, converting a one of a kind electronic system into a mass produced design, providing the level of support needed and expected by customers — these are not trivial tasks at all. One of the wonders of the field to this date is that the complicated objects of our computer affections are as free of design and assembly hassles as they are at the present time. Simply look at all the system design principles which have been properly implemented and provided in existing products, and for the moment ignore the obvious residual improvements that would make a complicated product "more perfect" by some standard.

But technology almost by definition is not static. One of the significant indications of personal computing's ever improving technology is the trend toward minimizing hassles involved in getting a production model computer system up and running. This trend affects both existing product lines and new products which will shortly come to market.

- *Item:* In talks with the MITS people at the recent World Altair Computer Convention, one point which was made is that an increasing number of customers order assembled and tested Altairs. This delivery of assembled units, both direct from the factory and through computer stores, significantly reduces the time and hassle overhead of getting a system up and running.
- *Item:* In a recent visit to Sphere, the same story was heard: While initially the kit orders predominated, a growing number of purchasers are opting for assembled systems.
- *Item:* A new firm (let it be known as "brand A" for now) recently contacted BYTE with a report on its product, the inspiration for this commentary. The product's significance is that it comes in one and only one form: a completely tested board of moderate size which needs only a power supply, monitor with EIA video input, keyboard with parallel TTL interface and audio cassette recorder to complete the system. The price of the board is well under $1000 and a BASIC software package is thrown in as part of the deal. The memory capacity of this 6502 based system was quoted as 8 K, with an option to replace the 4 K dynamic memory chips with pin compatible 16 K chips

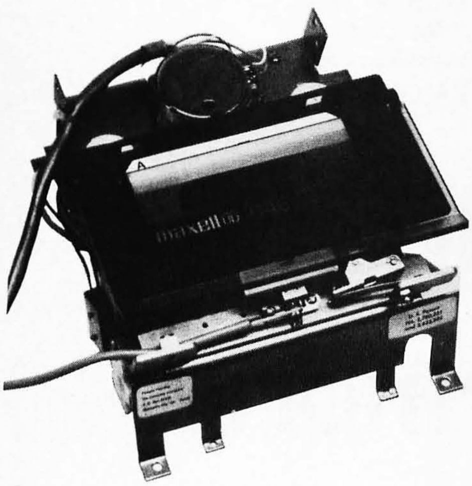
Continued on page 110

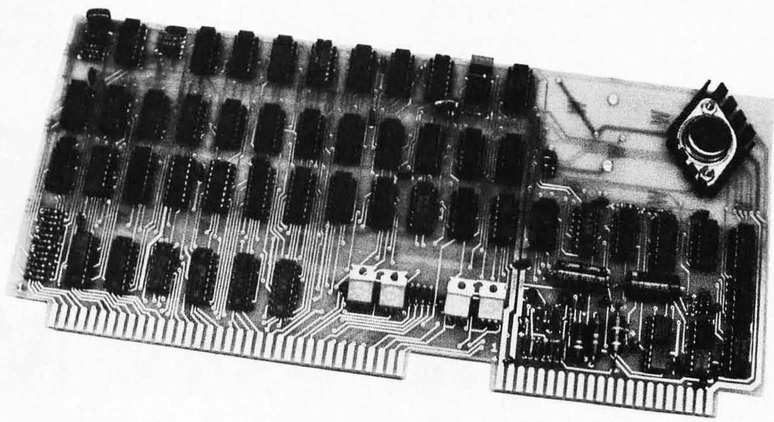
Articles Policy

BYTE is continually seeking quality manuscripts written by individuals who are applying personal systems, or who have knowledge which will prove useful to our readers. Manuscripts should have double spaced typewritten texts with wide margins. Numbering sequences should be maintained separately for figures, tables, photos and listings. Figures and tables should be provided on separate sheets of paper. Photos of technical subjects should be taken with uniform lighting, sharp focus and should be supplied in the form of clear glossy black and white prints (if you do not have access to quality photography, items to be photographed can be shipped to us in many cases). Computer listings should be supplied using the darkest ribbons possible on new (not recycled) blank white computer forms or bond paper. Where possible, we would like authors to supply a short statement about their background and experience.

Articles which are accepted are typically acknowledged with a binder check 4 to 8 weeks after receipt. Honorariums for articles are based upon the technical quality and suitability for BYTE's readership and are typically $15 to $30 per typeset magazine page. We recommend that authors record their name and address information redundantly on materials submitted, and that a return envelope with postage be supplied in the event the article is not accepted. ■

1,000,000 BYTES ON-LINE!





The Digital Group Cassette Storage System

The Digital Group Cassette Storage System gives you total magnetic tape data storage and retrieval for your microprocessor, capable of operating 1 to 4 computer-controlled Phi-Deck cassette transports. Within seconds (20 at most), your system zips to any of over one-quarter million 8-bit bytes per drive. And that really puts it all on-line!

The Digital Group Cassette Storage System is ideal for:

- Large data files – names, accounts, etc.
- Indexed computer-controlled program files
- Sorts
- Inexpensive mass storage
- Work files
- Indexed random retrieval
- Multi-pass compilers
- System residence

In addition, with a Digital Group System and a Phi-Deck transport, your total load procedure is reduced to a single action – turning on power. Everything else is automatic! Your Digital Group System is completely ready for use in a very few seconds. And you avoid a large investment in single-use PROM memory.

MAJOR STORAGE SYSTEM COMPONENTS

1. Controlling and Formatting Interface – single card for 1 to 4 drives
2. Software Operating System
3. Computer-controlled Cassette Drive(s)

Selected Specifications

Data Rate: 800 bytes per second, 8K loads in 10 seconds
Media: High-quality standard audio cassettes
Search Speed: 100 inches per second
Tape Speed: 5 inches per second
Power Requirements: +12V to +20V at .7A peak and +5V at 1A plus 60ma per drive
Port Requirements: One 8-bit parallel input port plus two 8-bit parallel output ports
Cassette Drive is an enhanced Phi-Deck with a digital head, cast head bar, stronger capstan, and four-foot cabling.

SOFTWARE OPERATING SYSTEM

8080 based – 650 bytes
Error Detection: CRC
Retries after soft errors
Automatically bypasses hard errors
Block size = 1 to 256 bytes or multiple of 256 bytes
Functions supplied:

- ○ Record multiple blocks
- ○ Record 1 block
- ○ Read 1 block
- ○ CRC check
- ○ Fast reverse
- ○ Fast forward
- ○ Search for block

For more information, drop us a line or call . . . but by all means, get on our mailing list.

Prices: Interface – full kit PHI-F . . . . $135 ppd
Each Drive – assem. PHI-1 . . . . $115 ppd

THE DIGITAL GROUP INC
P.O. BOX 6528
DENVER, CO 80206
(303)861-1686

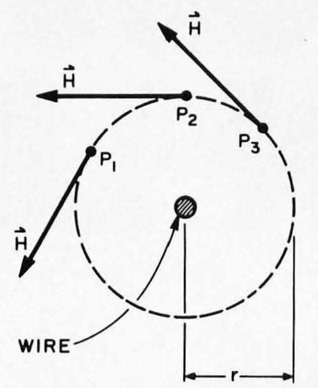


Figure 1: For a wire carrying current i directed out of the page, the magnetic field **H** *is proportional to i/r. The direction of* **H** *is tangential to the circle of radius r, as shown for points P_1, P_2 and P_3.*

Coincident Current Ferrite Core Memories

James R Jones
111 E Jefferson
Colorado Springs CO 80907

Have you ever looked through the surplus catalogs and wondered whether those memory core planes and stacks advertised could be used for anything other than tea strainers? How are they supposed to work, and what kind of circuitry is required? Could you troubleshoot and repair that surplus core memory and drive electronics? What good is a core stack without data? Is it a worthwhile project to build a memory system around a core stack?

Some of these questions can be answered directly in what follows. Others can be answered at least partially in terms of my own experience: getting one surplus core stack to work as the main memory of my home brew computer by building the necessary electronics without the benefit of manufacturer's data.

Perhaps the biggest advantage of the core memory over other types of random access memories is its ability to retain stored data when power is removed, and to have it readily available when power is restored. This feature is sometimes used by minicomputer manufacturers to ship their products with preloaded systems software. Also, for some forms of core memory, the storage capacity increases much faster than the amount of driving electronics. In order to understand the functional requirements and the operating restrictions placed on core memory driving circuitry, it is necessary to take a close look at how cores work, and how they are typically organized to form large capacity stacks.

Magnetic Fields and Hysteresis

The memory storage element is the ferrite core itself. Its function is to accept, store, and read out a bit of information. It can do this by virtue of its ferromagnetic properties. These allow it to be easily, but strongly, magnetized in a preferred direction by an externally applied magnetic field to signify a 1 bit value, to be easily magnetized in another direction to signify a 0 bit value, and to retain its magnetic direction when the external field is removed. Physically the core is a small doughnut shaped object made of pressed, heat treated, non-conductive, iron oxide powder. A typical dimension is 0.02 to 0.2 inches in diameter (0.5 to 5 mm). The external magnetic fields are applied to the core by means of wires passing through it carrying controlled amounts of current.

A straight wire carrying a current gives rise to a vector field, **H**, in the surrounding space, called the magnetic field strength. **H** is proportional to the current i, and is a vector directed as shown in figure 1. The arrows indicate the directions the north end of a compass needle would point if placed at the points shown. Another vector quantity, the

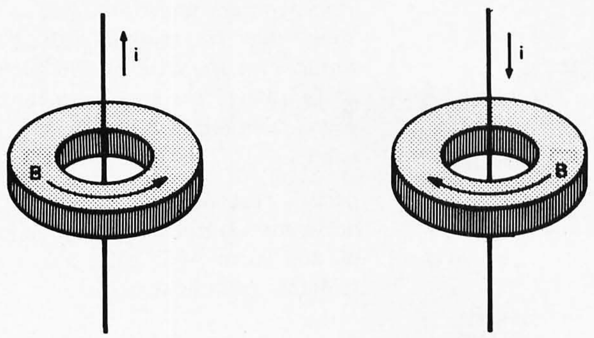


Figure 2: The direction of magnetic induction **B** *in the core results from alignment of core structures with the magnetic field due to current i.*

magnetic field of induction, **B**, is also considered to exist, and describes the overall magnetic effects due to **H** and the presence of matter. In figure 1, **B** behaves like **H**. If the wire passes through a ferrite core, the **H** field generated by the wire causes molecular and microscopic sized magnetic domains (regions magnetized like the compass needle) in the core to align themselves in the direction of **H**, resulting in a **B** field in the core with direction around the circumference of the core as shown in figure 2. This field is much stronger than that due to the wire alone, because the structures in the core add their fields to the wire field when aligned.

Because of the geometry of the cores with respect to the wire, it is possible for us to drop the vector notation and simply refer to the magnitude of **B** and **H** in the discussion which follows. The exact way the strength of **B** in the core depends on the strength of applied field **H** (proportional to current i) is represented graphically in the scalar B versus H curve of figure 3. After manufacture, the core has no magnetization, and B = 0. If applied field H is increased from 0 to Hp, B increases along path 0-l-c-e., If H decreases to 0, B decreases from e to b, leaving the core magnetized with no applied field present. Decreasing H from 0 to Hm causes B to move along b-a-g-f, changing direction as it passes through 0. As H increases to 0, B moves to h, again leaving the core magnetized with no applied field, but in a direction opposite that above. As H increases again from 0 to Hp, B moves along h-j-d-e. If H is again varied as described above, B will trace the same counterclockwise path.

This effect of tracing two alternate paths between two fixed states, instead of tracing a single path, is called hysteresis. In the core,

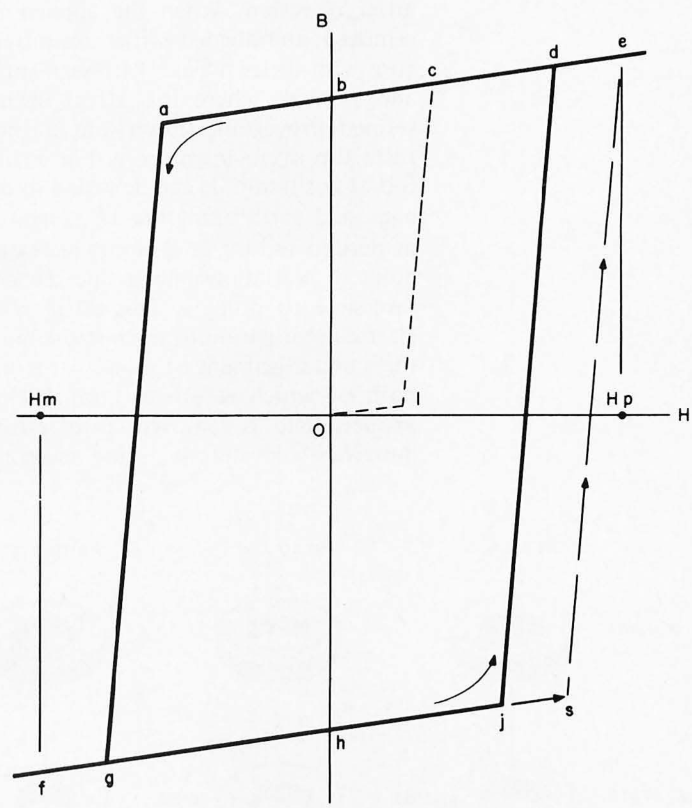


Figure 3: Memory properties are derived from the hysteresis properties of the core material. When the applied H value is 0, the residual B value in the core material will depend upon the past history of magnetization. This figure and subsequent figures show the magnitude of the vector quantities **H** *and* **B**, *with positive and negative values corresponding to the two directions of rotation about the core axis.*

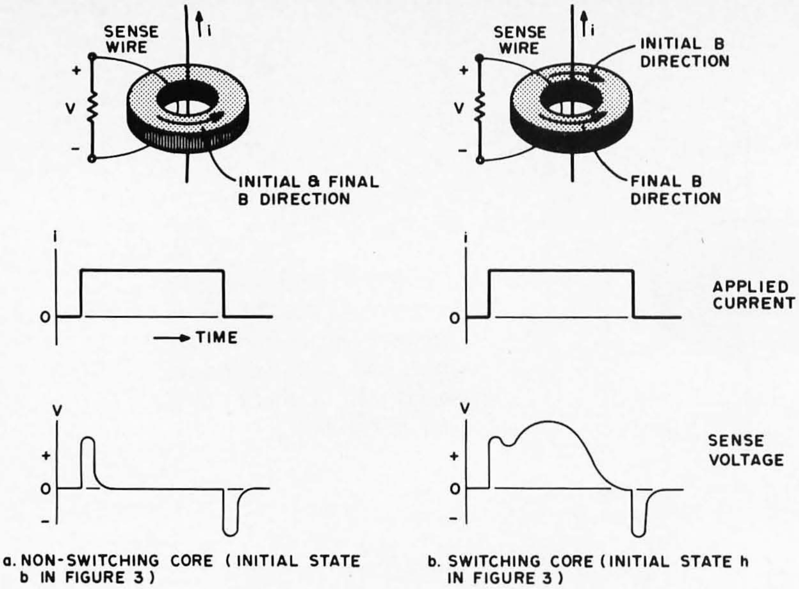


Figure 4: Sense voltages perceived on a wire strung through the core depend upon the previous **B** *magnetization state in the core as a current pulse is passed through the wire. A non-switching core produces a simple transient due to reversible* **B** *field changes induced in the core by the current edges as at (a). When the core switches as at (b), an extra transient pulse is induced in the sense wire.*

this is due to the alignment of the magnetic domains, a process which requires a minimum applied field to force alignment in a given direction. When the applied field is removed, the aligned structures provide the core with its own field. Path segments a to g and j to d, where this effect occurs, are termed irreversible. Reversible magnetic effects also occur in the core. For example, if B is at position h, H can be varied to move B back and forth along line f-j as many times as desired as long as B is not increased past point j. Variations along line a-e are also reversible to point a. This effect is due to elastic deformation of the microscopic structures and alignment of molecular structures, both of which return to normal when the applied field is removed. Both effects are important: Hysteresis is the memory property; reversibility, although the largest noise contributor in the core, allows the core to act as an AND gate, a key principle in selection of individual bits.

Sensing Core Magnetism Changes

Changes in the core B field are sensed by means of a second wire passed through the core. The voltage difference appearing at the ends of the sense wire is proportional to the change in B, and inversely proportional to the time required for the change. So that an irreversible change can produce a large signal, H is made to change as fast as possible. However, this causes the reversible changes to produce large signals also; but the speed tends to separate the signals: Deformation effects can occur much more rapidly than gross microscopic realignments, eg: for a fast increase of H from 0 to Hp in figure 3, B would not follow h-j-d-e, but rather h-j-s-e, so most reversible changes occur first. Figure 4 illustrates the voltages appearing across the sense wire for a core in initial state b, and in initial state h, resulting from a fast rising current pulse generating field Hp.

Using the Core to Remember

If we now interpret the initial direction of **B** in the cores to represent a stored 0

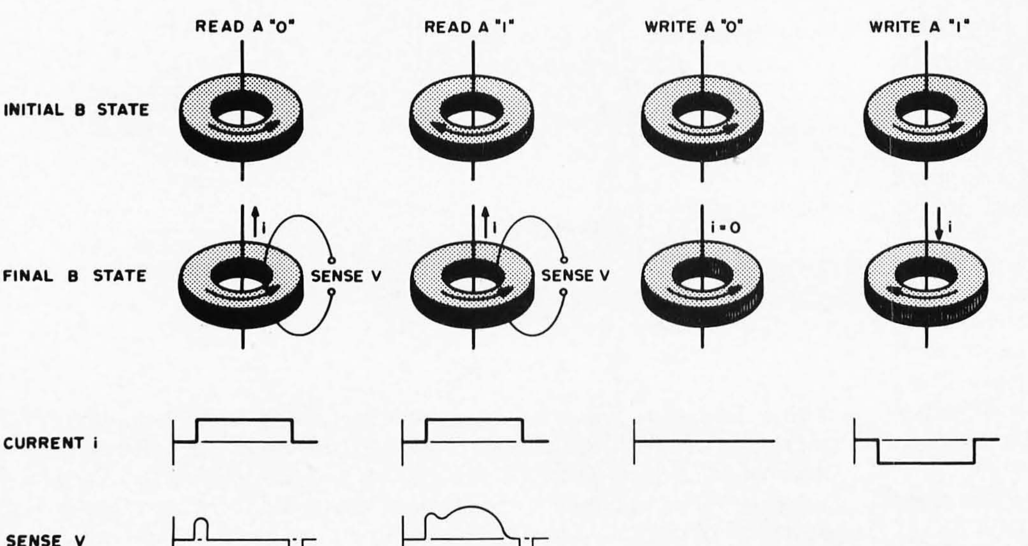


Figure 5: Using the physics of core switching. This figure shows the four basic operations needed to manipulate the binary state of a memory core: Read a "0", read a "1", write a "0" and write a "1".

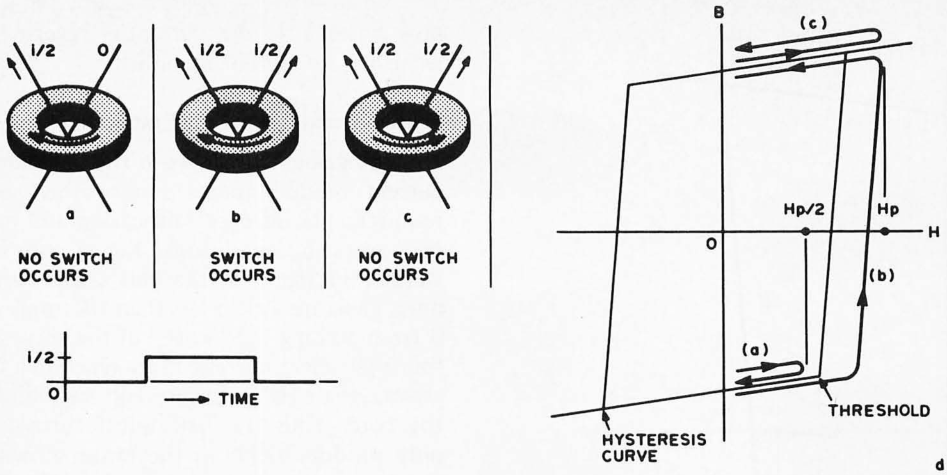


Figure 6: How hysteresis makes coincident current bit selection possible. The principle involved is simple: A current of i is sufficient to generate an H field strength which will drive the core from one state to the opposite state, but a current of i/2 is insufficient to cause the core to cross the threshold of the hysteresis curve. The H field of two separate wires add so that if two wires carry currents of i/2 in the same direction, the result is the same as a single wire carrying a current of i. The hysteresis curve at the right (d) shows the paths taken by the B field in the case of a single i/2 current (a), additive i/2 currents forcing a change (b), and additive i/2 currents causing no change since the core is already in a state aligned with the H field (c).

(figure 4a) or a stored 1 (figure 4b), and read the core contents by the method of figure 4, then read and write operations can be performed as indicated in figure 5. Two things should be noted: First, a stored 1 is destroyed in the process of reading it because the core is reset to the 0 state. If it is necessary to retain the 1 in memory for later use, a special refresh write operation is required to restore the 1. Second, the cases of writing into a core already containing a 1 are not shown, because the memory control logic is usually designed such that a destructive read cycle always occurs before the write cycle, for a given core.

Memories Larger Than One Bit

Using the method of figure 5 can be an expensive undertaking if one wants to store 4096 bits, for instance. Although the sense line can be made to pass through all the cores to detect switching of a selected core, circuitry must be built which can select any one of 4096 cores and supply the proper read or write current to it alone. To reduce the amount of selection circuitry required for large memories, the cores' reversibility property is used to assist in the selection process. This is done by passing two independently controlled current lines instead of just one through a given core; each line can pass half the current of the original line. Due to the shape of the B versus H curve, if only one line passes half the required switching current required, B remains in the reversible region and returns to the original state (figure 6a) when the current is removed. If both lines pass current, core switching occurs if the core is in the proper original state (figure 6b and c).

Coincidence Current Selection

Thus by the means of coincident currents, the core acts as an AND gate; both currents need to be present simultaneously

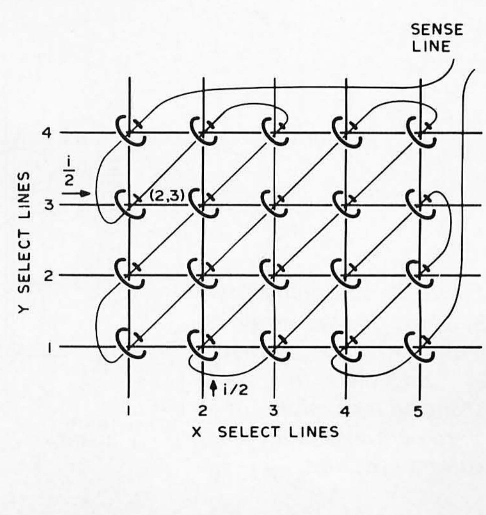


Figure 7: The concept of a coincident current selection is implemented using an array of cores with X and Y lines. In this example, a total of 20 cores is wired with a single sense wire, 4 Y selection wires and 5 X selection wires. When i/2 (see figure 6) is flowing through one X and one Y line, the matrix intersection point is addressed. In this example, the core at (2,3) will either be read or written. Actual core arrays are much larger than this simple conceptual illustration.

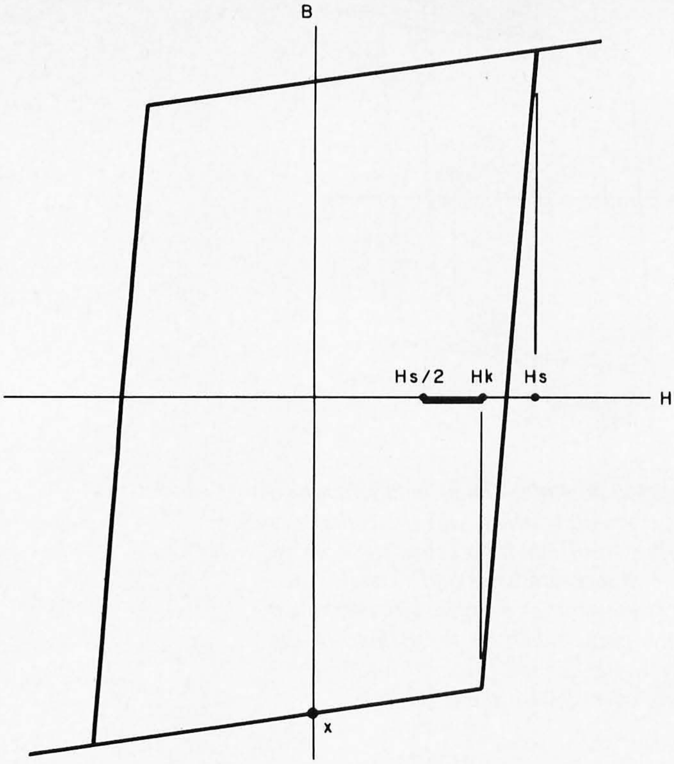


Figure 8: The half-select current i/2 must generate an H field less than Hk if a change of state is to be avoided. Similarly, the full select current of i must generate an H field strength greater than Hs to ensure switching. This leaves a range of field strengths (and corresponding current values) from Hs/2 to Hk which introduces a critical tolerance for the select currents in a core memory design.

to choose the core and allow it to switch if it is in the proper state. A multiple bit memory can now be constructed as shown in figure 7. If currents are applied to lines X2 and Y3, the core at (2,3) is selected. Other cores on the lines X2 and Y3 are half selected. Only one pair of lines is allowed to carry current at any one time. To read the core contents at (2,3), current is passed as indicated. To now write a 1, the current is reversed; to write a 0, no current is applied.

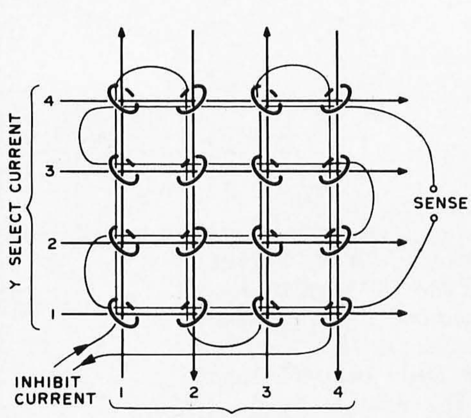


Figure 9: The inhibit line is added to the array assembly of cores so that it is possible to avoid changing the state of a given core with coincident current selection.

Property and Operation Restrictions

Arrangement of cores in the coincidence current mode imposes some rather severe requirements on core fabrication and memory operating conditions. For a core in B state x in figure 8, the half select current must generate a field less than Hk to prevent B from passing the "knee" of the curve. But the total select current must generate a field greater than Hs to ensure full switching of the core. Thus the half select current can only produce fields in the range shown by the heavy line. As the quantity (Hs – Hk) gets smaller, the operating range gets larger. So the cores are made with (Hs – Hk) as small as possible, but even if it were zero, the half select current would have a maximum tolerance of 33% of its midrange value. The operating range is further restricted by two more factors. First, manufacturing and assembly tolerances allow memories to be constructed of cores with slightly differing B versus H curves, effectively increasing the difference (Hs – Hk) as all cores are considered. Second, as the memory is operated, the cores heat up due to switching losses, select line heat, and heat from nearby electronics, causing the entire B versus H curve to shrink towards the origin. Newer core memories are made using ferrite with low temperature sensitivity; but this is not true of older memories, and such measures as constant temperature ovens, forced air cooling, and power supply temperature compensation have been used to ensure reliable operation.

Reading and Writing Words: The 3D Memory

The coincidence current scheme certainly cuts down the required selection circuitry, although at the expense of operating tolerances. The 20 cores of figure 7 are selected by nine lines, and the 4096 bit memory mentioned earlier, if arranged in a 64 X 64 array, has only 128 select lines. Further economy is realized if the idea is expanded to word organized memories. For example, consider the 16 bit array of figure 9 which has an added winding – the inhibit line (the arrows on the select lines indicate the read current directions). If a 16 word memory of 3 bits per word is desired, three of these bit planes are interconnected as shown in figure 10.

Sending half select currents through X1 and Y1 select lines in the arrow directions will cause the core at (1,1) in each array to read out via the associated array sense line. Reversing the select current directions will

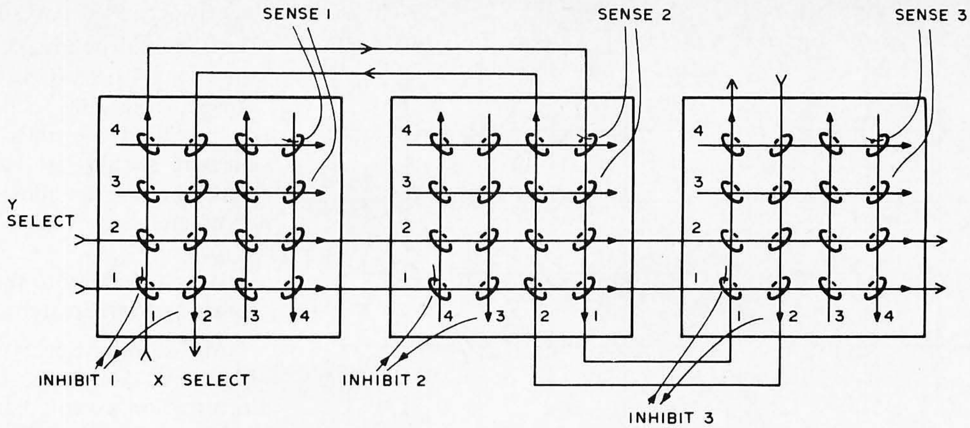


Figure 10: 3D Plane Interconnections. In a 3D memory, three dimensions of addressing are implicit: There is an X and Y dimension for each plane, and one plane is used for each bit of the parallel computer word. In this picture, three planes of a 3 bit wide memory are shown. Only two lines are shown connected in each X and Y direction. Other X and Y lines are connected between planes in a similar fashion.

cause a 1 to be written in each (1,1) position. In order to write a 0 in some of the (1,1) positions, a half select current is also applied to the desired inhibit lines in the direction of the arrows at the time 1's are to be written. This inhibit current appears as read half select current at all cores of the associated array, cancelling half the write select current at the selected core, and thus preventing the writing of a 1 at that core.

For a 4 K word memory of 16 bit words, the assembly (called a stack since the arrays are often stacked like pancakes for simple interconnection along the edges) would have 128 select lines, 16 sense lines, and 16 inhibit lines. The connection scheme is termed 3D, and is most economical in terms of support circuitry.

Other Arrangements

Bit oriented memories of the form of figure 7 are called 2D memories. Another commonly used word oriented connection scheme, 2½D, will not be discussed because of the relatively large amount of select circuitry used. Interested readers are referred to the bibliography at the end of this article. A variation of the 3D memory replaces the inhibit and sense lines by a single line which performs both functions, since they occur at separate times. At the cost of additional circuit complexity, the variation allows smaller cores and more closely packed arrays to be used.

Minimizing the Noise

A core can be oriented two ways at a select line intersection, and sense and inhibit lines can string the cores of a plane in many patterns. Advantage is taken of these facts to minimize noise from sources which could otherwise mask the sense line signal due to core switching. Major noise contributors are pulses from half selected cores (path a in figure 6d). Although small for a single core, the composite signal due to 126 half selected cores in a 4 K array will spread and totally mask a switching signal. A diagonal pattern that results in almost complete cancellation of half select pulses is shown in figure 11a. Noise is also generated through capacitive coupling between select and sense lines, allowing select current edges to induce ringing in the sense line. This effect can be minimized by running the sense line parallel to one select line direction as shown in figure 11b. The parallel select line is turned on first, a large noise pulse is induced which dies down, then the other line is turned on to select the core. Little noise is now generated because capacitive coupling is small.

There are other lesser noise effects which make the 4096 core array about the largest to be practically served by a single sense line. Larger stack word capacities are realized by using multiple sense lines on each plane, thus requiring additional sense circuitry.

Driving the 3D Memory

The half select current tolerances mentioned in connection with figure 8 are further reduced to less than 20% per line by the addition of the inhibit line. Unfortunately, the amplitude of the core switch signal depends on the total select current

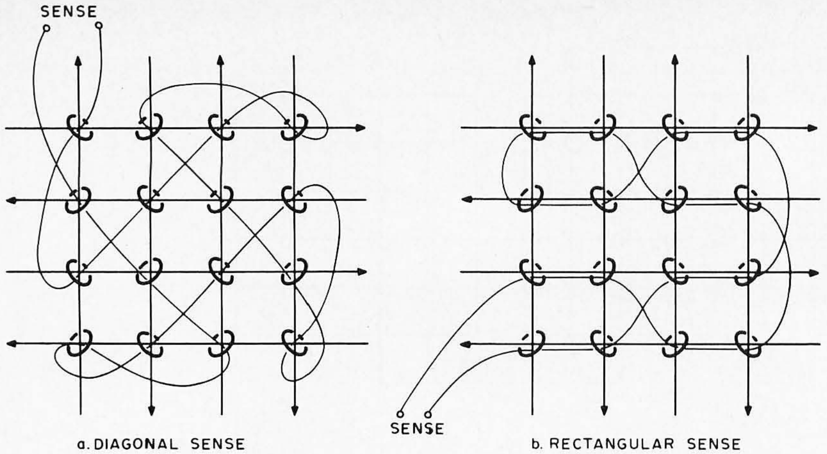


Figure 11: Noise pickup is minimized by threading the sense line through the core array in a way which causes induced noise voltages to cancel each other.

present, so the select current must be maintained at the higher allowed values to obtain a favorable signal to noise ratio on the sense line. This reduces current tolerances to around 5% to 10%. Depending on the type of cores used, half select currents are in the range of 150 to 500 mA. Current rise time of the last select line turned on (to read) should be about half the core design time for a 1 peak (figure 4b) to occur, which ranges from 100 nanoseconds to about a microsecond for older large diameter cores. Current should last for at least twice the peaking time to allow the core to switch completely.

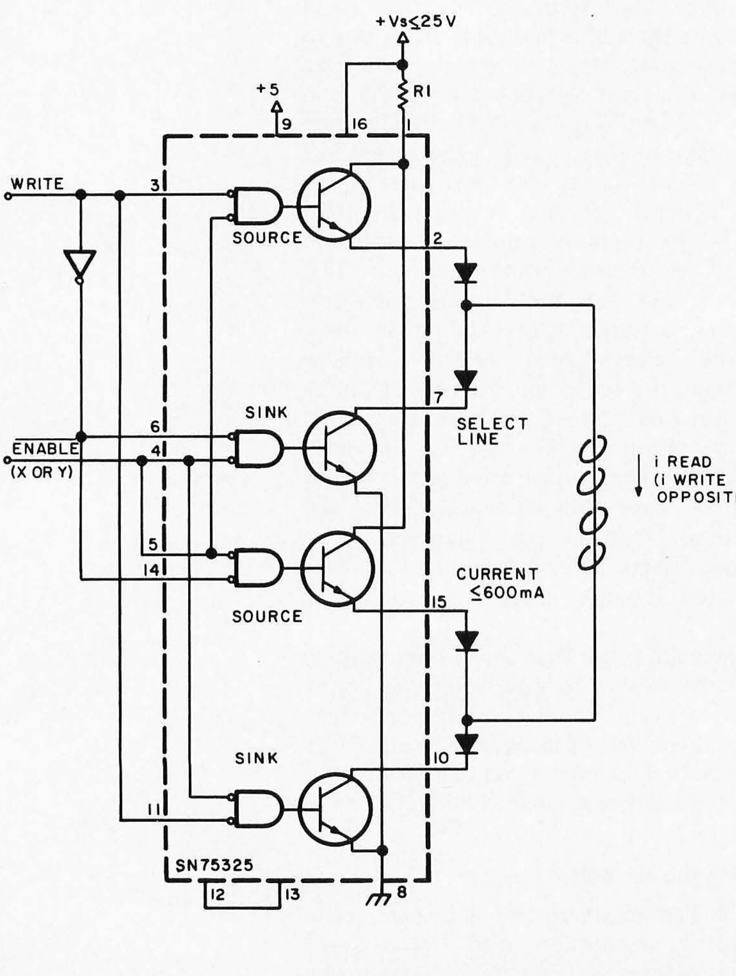


Figure 12: Select Line Test Driver. For testing, build two of these with one used to drive an arbitrary X select line, one used to drive an arbitrary Y select line. R1 is a current limiting resistor selected as described in the text.

Getting a Memory to Work (Starting from Scratch)

My surplus Spectra 70 stack (built by Electronic Memories, Inc) arrived with no information except that its organization was 128 x 136 words x 18 bits with four sense line groups of 64 x 68 words. Of that I intended to initially use a 64 x 64 word x 16 bit segment. Not being able to obtain manufacturer's data, I proceeded to discover the stack characteristics for myself. For readers in similar situations, I would recommend such a course of action only if you have lots of spare time, like surprises, and have access to a 15 MHz dual trace scope.

The physical dimensions of the stack were 5 by 5 by 3 inches (12.7 by 12.7 by 7.6 cm) not counting connectors. Tamper proof, auto destruct construction precluded a view of cores or windings. X and Y lines were labelled; an ohmmeter was used to associate sense and inhibit connector pins to core planes. Memory construction suggested the rectangular winding of figure 11b.

Tests

Several test circuits were built to determine core switching characteristics. Figure 12 shows the select line driver circuit, two of which are required to drive an X and a Y line. The Texas Instruments Memory Core Driver integrated circuit (SN75325) is used. The external diodes simulate the steering diodes required to select more than one line, and to direct reverse voltages (generated at current turn off time by the inductive load of the select lines) to the sink transistor collector which is internally diode clamped to source voltage Vs. Vs is regulated and variable over about a 5 volts range, and R1 is chosen for the current range to be tried. Figure 13 shows the inhibit driver circuit. The dual gates of the integrated circuit are connected in parallel to increase the current sinking capacity. Again the output must be diode clamped, and R2 is chosen as required. Figure 14a shows the sense amplifier with terminating resistors and means for varying the threshold voltage. The sense amplifier operates by generating an internal voltage proportional to the voltage difference appearing on the sense line wire pair. This voltage is compared with the adjustable

threshold reference voltage, and a TTL high level output is generated if the threshold is exceeded. Figure 14b illustrates the outputs produced at two threshold levels for an arbitrary input signal.

Timing signals used are shown in figure 15 along with resulting current directions. The logic diagram illustrates a circuit which can be used to generate this timing. The enveloping of X ENABLE and Y ENABLE eliminates part of the half select noise in the read cycle, and ensures that the inhibit current is at maximum level when full write selection takes place. Repetition rate of the memory cycle during testing is kept under 1 kHz to prevent damage to the SN75325s. Changing the mode, WRITE, with both enable signals low ensures that a glitch will not short current through a directly connected source to sink pair.

Two select line drivers were connected to an arbitrary select line pair, and the sense amplifier was connected to a sense line thought to pass through the X Y intersection. With no signals to the select line drivers, the sense amplifier threshold voltage was advanced from zero until SENSE OUT was a steady logic 0. Starting with a low (about 150 mA) half select current, the line drivers were started, and, with the sense amplifier output monitored on the oscilloscope, the threshold voltage was adjusted to show a series of spikes corresponding to select current edges as shown in figure 16a. As half select currents were increased equally, additional pulses emerged after spikes numbered 3 and 11, as indicated by the arrows in figure 16a. These pulses indicated core switching. Some experimenting was required to obtain proper select current balance and strength, which occurs when the switching pulses completely disappear (and not just shift) when either half select current is temporarily interrupted. Too much drive current was also indicated by a decrease in size of the switching pulses with corresponding pulses appearing behind spikes number 1 and 9.

Choosing inhibit drive resistor R2 to provide half select current in the range determined necessary above, the inhibit line corresponding to the sense line used was found by driving an inhibit current through each inhibit line in turn until the switching pulses were seriously disturbed. Further adjustments of the inhibit, X, and Y currents eventually resulted in the switching pulse appearing and disappearing as the inhibit current is turned off and on. In my memory stack, about 380 to 400 mA half select current is required to cause the core to switch, producing a sense amplifier output

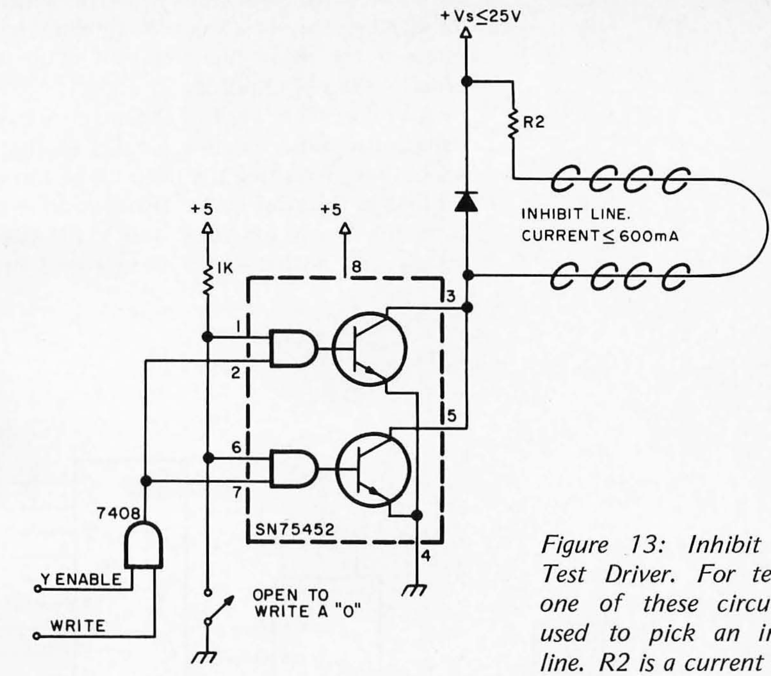


Figure 13: Inhibit Line Test Driver. For testing, one of these circuits is used to pick an inhibit line. R2 is a current limiting resistor set as described in the text of the article.

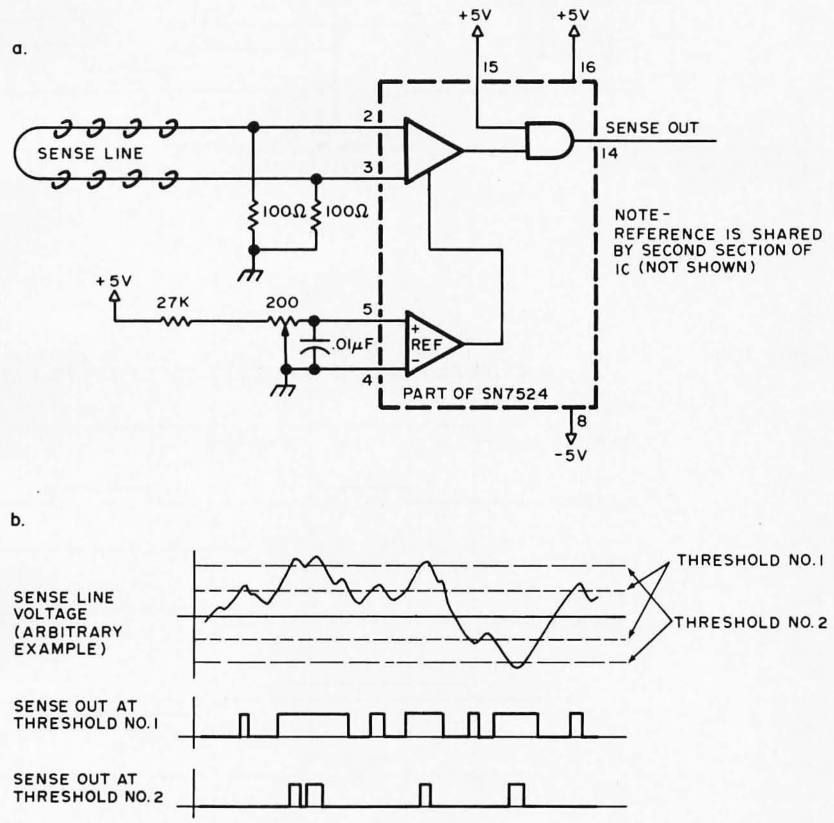


Figure 14: Sense Amplifier. The purpose of the sense amplifier is to create an output signal which can be strobed into a TTL latch at the appropriate moment, given the signals induced into the sense wire of the memory. For testing, the thresholds are set using the variable resistance of 200 ohms in a voltage divider from the +5 volt supply.

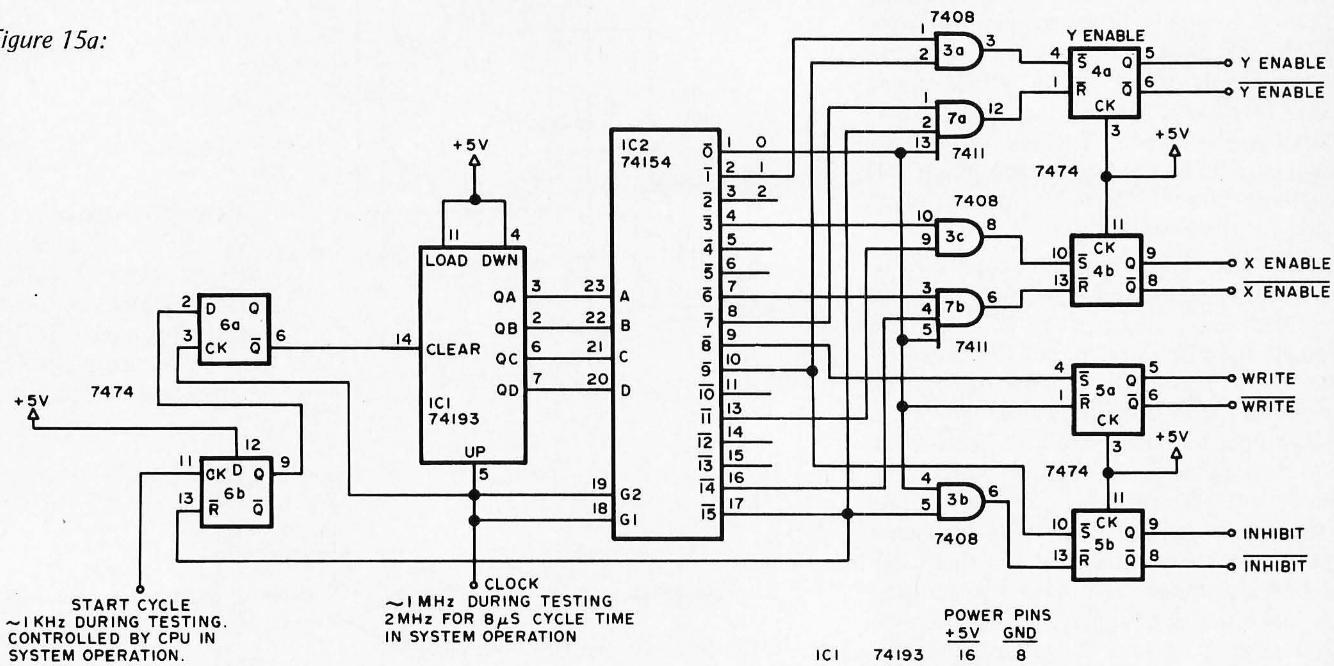
pulse sketched in figure 16b. The smaller pulse generated for a 0 is also shown. The 1 pulse at the sense line terminals is on the order of 40 mV amplitude.

An experiment verified the memory has a rectangular sense winding parallel to the Y select lines, requiring the latter to be turned on first in the read cycle. This was done by swapping the X ENABLE and Y ENABLE signals, and noting which arrangement produced the most sense noise at clock time 3 of figure 15.

Selecting the Select Lines

Success in getting the stack to respond to prodding induced me to go ahead with memory circuit construction. The SN75325s were employed in a commonly used scheme of line selection illustrated in figure 17, in which one of four X lines is chosen by four source to sink pairs, determined by a two bit

Figure 15a:



		POWER PINS +5V	GND
IC1	74193	16	8
IC2	74154	24	12
IC3	7408	14	7
IC4	7474	14	7
IC5	7474	14	7
IC6	7474	14	7
IC7	7411	14	7

Figure 15b:

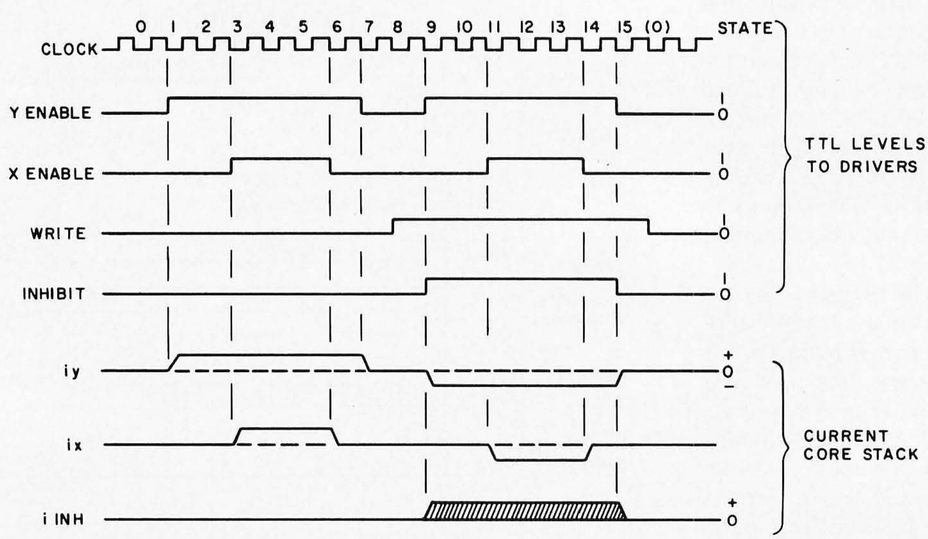


Figure 15: A suggested circuit (a) and timing wave forms (b) for driving core memories in testing and later in the final working version. The cycle time is set by the clock into pin 5 of the counter IC1. A positive transition at pin 11 of IC6 initiates a single memory cycle illustrated in (b). During testing, approximately 1 kHz will provide a good repetition rate which avoids burning up driver circuits; in final system operation, this circuit can also be used to generate the read then write cycle of core memory operation. For an 8 μs full cycle time, the clock should be 2 MHz.

address (low order bits 0 and 1). X READ ENABLE equals X ENABLE AND NOT WRITE; X WRITE ENABLE equals X ENABLE AND WRITE. By extension of this scheme (see the TI catalog listed in the bibliography), 16 source to sink pairs are required to select one of 64 X lines. I found that a single source resistor (R1 of figure 12) could service all source transistors used to select X drive lines, which eliminated the necessity of matching resistor values to provide the same current flow as different lines are selected. Two 3 to 8 decoders use six address bits to enable two source to sink pairs. The above circuitry is duplicated for the 64 Y lines, thus a 12 bit address is decoded into a unique X Y intersection of the core matrix. A total of 16 SN75325 integrated circuits and 4 decoders are required, along with diode arrays. Three independently regulated and variable supply voltages were provided to allow adjustment of X, Y, and inhibit currents. Inhibit drive resistors (R2 of figure 13) must be closely matched to allow operation from a common supply voltage.

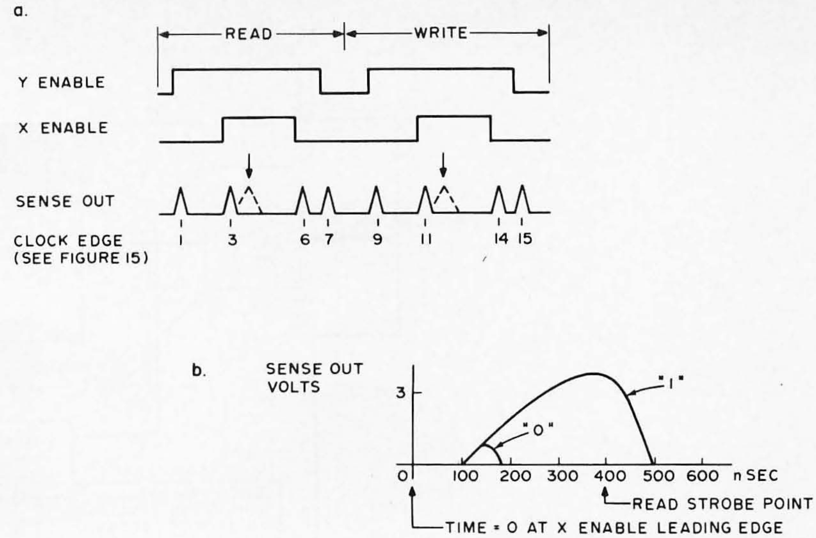


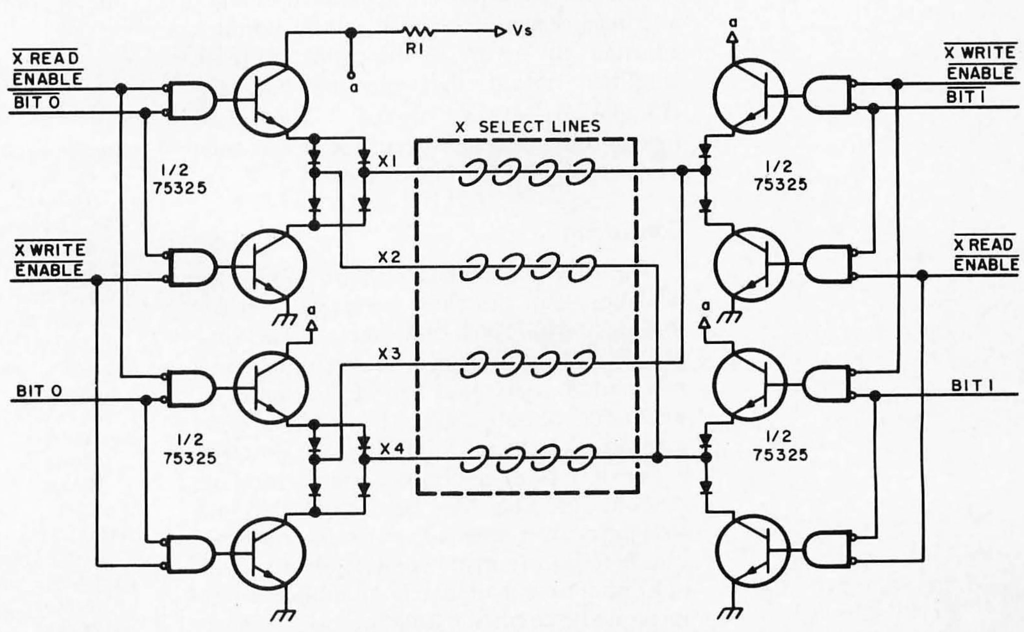
Figure 16: Sense Amplifier Outputs. Spikes will be seen in the sense amplifier outputs following current changes through the cores. Detection of a "0" or "1" is obtained by observing at the "read strobe point" which follows the leading edge of the X enable pulse. This observation point is typically 400 nanoseconds after the leading edge of the X enable pulse which drives one of the X select lines.

Memory Operation

Figure 18 illustrates control and data flow of the memory logic. Read data is stored, by means of the Read Strobe signal, in a buffer where it is used for restoring the memory word in a central processor read operation. In a central processor write operation, this read information is ignored and data from the CPU controls the inhibit drivers during memory writing. Timing is of the form of figure 15, except that the memory cycle (read then write) is accelerated to 8 microseconds in my system. Inhibit and sense circuits are similar to those of figures 13 and 14a.

The arrangement of figure 18 is wasteful of central processor time, because data and address information must be held until the memory cycle is completed. Buffering data and address information in the memory logic would allow the processor to continue after

Figure 17: X Line Select Switch. This arrangement is an extension of that illustrated for testing purposes in figure 12. For 75325 pin connections see figure 12, or page 10-21 of The Linear and Interface Circuits Data Book for Design Engineers *by Texas Instruments, 1973 edition.*



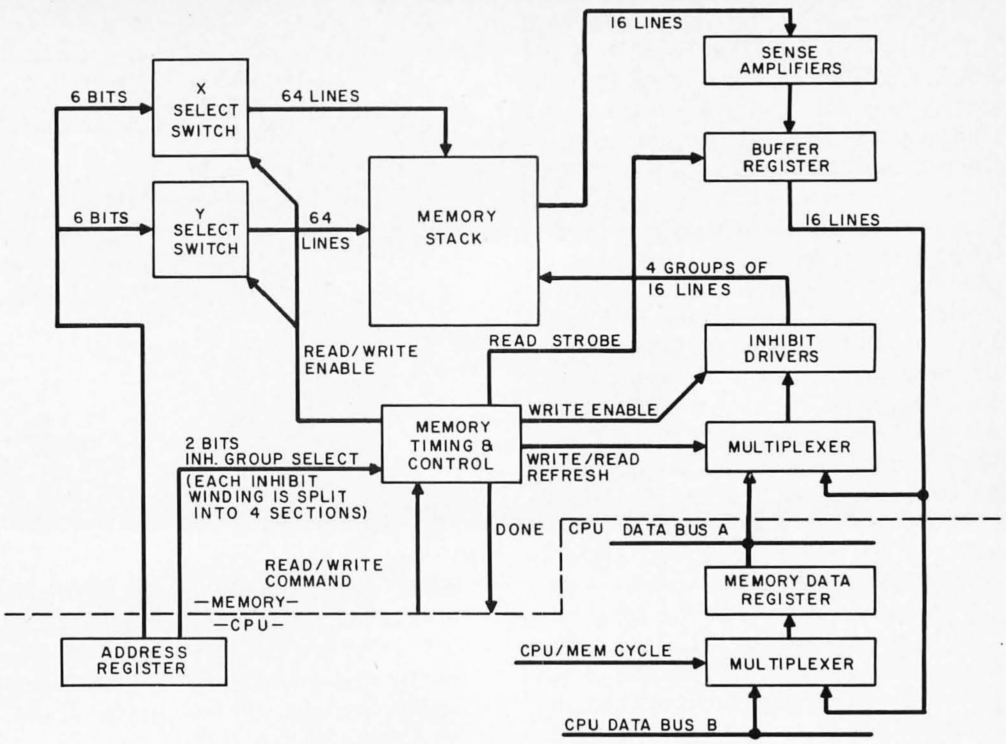


Figure 18: System diagram of a memory built according to the principles described in this article. The basic memory cycle is set up by a circuit such as the one shown in figure 15 as a source of timing control. In a read cycle, data is first read, then rewritten; in a write cycle, old data is read and new data written.

issuing a write command, or to wait for only half the memory cycle for data needed by a read operation.

The memory is coarsely adjusted by electronically forcing repeated manual write cycles through the processor's control panel. This allows adjustment of the half select and inhibit currents, sense amplifier thresholds, and read strobe. The read strobe point is adjusted to occur at the point of sense amplifier output that provides best discrimination between 0 and 1 levels (see figure 16b). Short program loops can then be used to make finer adjustments.

Conclusion

Recently completed, the 4096 x 16 memory subsystem described here was built for less than $400, including stack, power supplies, and miscellaneous electronic and mechanical parts; and can be expanded to 16 K for about $200. In most of the memory, storage reliability is good; although a few blocks of addresses tend to pick up random bits. Bugs have been eliminated in a straightforward manner, most having been traced to poorly made connections to select lines and to a bad batch of inhibit driver integrated circuits. Building a memory around a core stack is an interesting but time consuming project that can definitely result in a usable end product. As with many surplus items however, performance depends a lot on how much you can find out about the stack, and on the hardware design concessions that have to be made in the interest of financial solvency.■

BIBLIOGRAPHY

Fundamental Principles of Switching Circuits and Systems. New York: American Telephone and Telegraph Company, 1961.

The Integrated Circuits Catalog for Design Engineers. Dallas: Texas Instruments, Inc (First Edition).

Daly, Edmund B: "Noise Problems in Ferrite Core Coincident Current Memories." *Computer Design* (November 1967), 46-50.

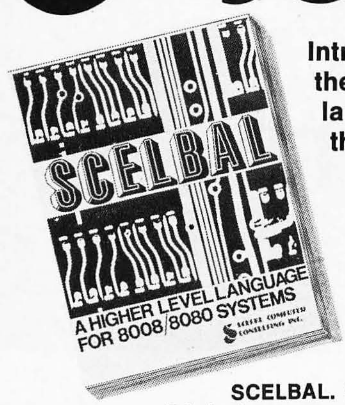
Hawkins, Joseph K: *Circuit Design of Digital Computers.* New York: John Wiley and Sons, Inc, 1968.

Kirkpatrick, T, and D Johnson: "Design of Modular, Low-Cost Core Memories for SLT Systems." *Computer Design* (April 1968), 60-71.

Smith, James E: "A 650 Nanosecond 2½D Memory." *Computer Design* (October 1966), 80-87.

Zinschlag, Howard P: "A 2½D Integrated Circuit Memory." *Computer Design* (September 1966), 26-39.

SCELBI SOFTWARE



Introducing SCELBAL, the new microcomputer language that's simpler than machine language.

A complete language for "8008"/"8080" systems including source listings, routines, flow charts and more!

SCELBAL. SCientific **EL**ementary **BA**sic **L**anguage for "8008"/"8080" systems. A complete, illustrated program book. Routines. Techniques. Source Listings. Flow Charts. And more. Took several years to develop. Now yours for many years to come. First time that intimate details of higher level language has been offered for such a low price. Only $49! You get **5 Commands:** SCR, LIST, RUN, SAVE, LOAD. **14 Statements:** REM, LET, IF . . . THEN, GOTO, FOR with STEP, END, INPUT, PRINT, NEXT, GOSUB, RETURN and optional DIM. **7 Functions:** INT, SGN, ABS, SQR, RND, CHR, TAB. And, it runs in 8K and more. Here's all the data needed to customize a high level language for your system . . . at a fraction of the cost!

Order your copy today! Get started advancing your system! $49 ppd.



Here's SCELBI's First Book of Computer Games for the 8008/8080. Action-packed. And fun.

Try to beat the computer at its own game. Here's the first complete machine language computer manual for computer games to include source listings, flow charts, routines and more. **Space Capture** — You against the computer using "search and destroy" strategy to shoot down roaming alien spaceships in outerspace. **Hexpawn** — a mini-chess game that lets the computer make mistakes . . . but only once. **Hangman** — an updated version of the great kid game. Computer selects words at random from long, expandable list. Try to beat it in 8 moves or less. Illustrated. Fun extras to put your computer to challenging, competitive, fun use. **Order yours today! 14.95 ppd.**

SCELBI's new microcomputer Game Book of Outer Space . . . GALAXY

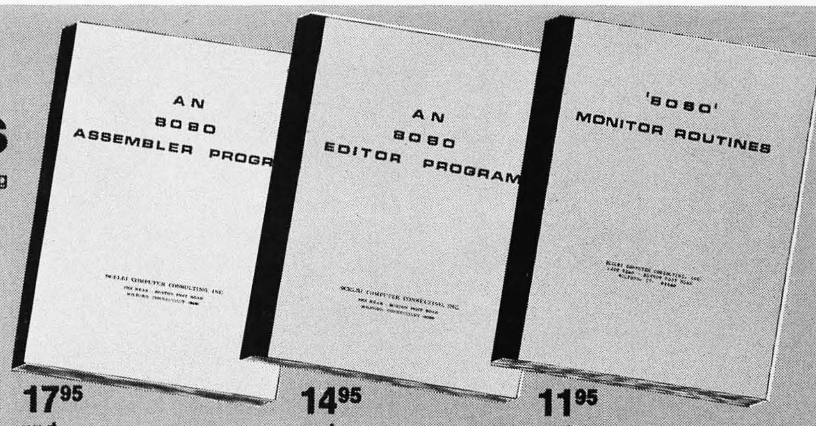
Captain your own crusading starship against the logic of your "8008" or "8080". You must destroy a random number of alien ships without running out of stardates, out of fuel, out of ammunition or out of the galaxy. Plan your mission. How much fuel is used for your warp factor? Don't run into roaming stars that can damage your ship. **Suddenly! Condition Red! Alien in sight!** How big is he? Fire a phasor or torpedo! He's damaged or destroyed. But, you've used valuable fuel. That's just the beginning. A new game every time. Complete program book in machine language for 4K memory, including source listings, flow charts, routines, etc. Great intergalactic adventure and fun.



Order "GALAXY" today! 14.95 ppd.

"8080" SOFTWARE MANUALS

Three new basic, complete "MUST" manuals! **"8080" Assembler Program** operates easily in 4K bytes of RAM (including symbol table). Unique feature: How to accept mnemonics related to "8008" based machines on "8080". Includes all major routines, pertinent flow charts, highly commented assembled listing and more. **$17.95.** **"8080" Editor Program** identical to "8008" Editor, with machine codes for "8080". **$14.95.** **"8080" Monitor Routines** same functions as "8008" specifically developed to utilize expanded capabilities of "8080". **$11.95.**



17.95 ppd. 14.95 ppd. 11.95 ppd.

SAVE $5.35! BUY ALL 3 FOR ONLY $39.50!



19.95 ppd.

MACHINE LANGUAGE PROGRAMMING FOR THE "8008"

(and similar microcomputers)

Here's the detailed, basic manual you need to develop today's machine language programs. 170 pages. Illustrated. Easy-to-read, understand. Most techniques applicable to other micro-computers, including "8080". Floating-point arithmetic package is worth the price alone! Order today!

- Detailed presentation of "8008" codes • Flow Charts
- Mapping • Floating-point Package • Debugging
- Basic programs: loops, counters, masks
- Organizing Tables • Editing/Assembling
- Math operations • I/O, Real Time Programming
- Maximizing memories • And lots more.

Master Charge, Postal and Bank Money Orders preferred. Personal checks delay shipping up to 4 weeks. Pricing, specifications, availability subject to change without notice. Prices for U.S. and Canadian delivery at book mailing rate. Add $2.50 for each publication if Priority Air Service (U.S.) desired. Foreign orders add $6.00 for each publication.

SCELBI COMPUTER CONSULTING INC.

1322 Rear Boston Post Rd., Milford, CT 06460
Telephone: 203/874-1573

Assembling a Sphere

Bruce A Anderson
4554 Chinook Ct
San Diego CA 92117

Sometime in July an advertisement was being run in several magazines which offered a complete computer system for only $650. Up until this time I had been interested in microprocessors but had not even remotely considered buying one. I sent off for the brochure anyway. After reading the claims about the system, I quickly convinced myself that I could justify the expense as an educational expense; so I sent off a check to Bountiful UT where resides Sphere Corporation, the company making these claims.

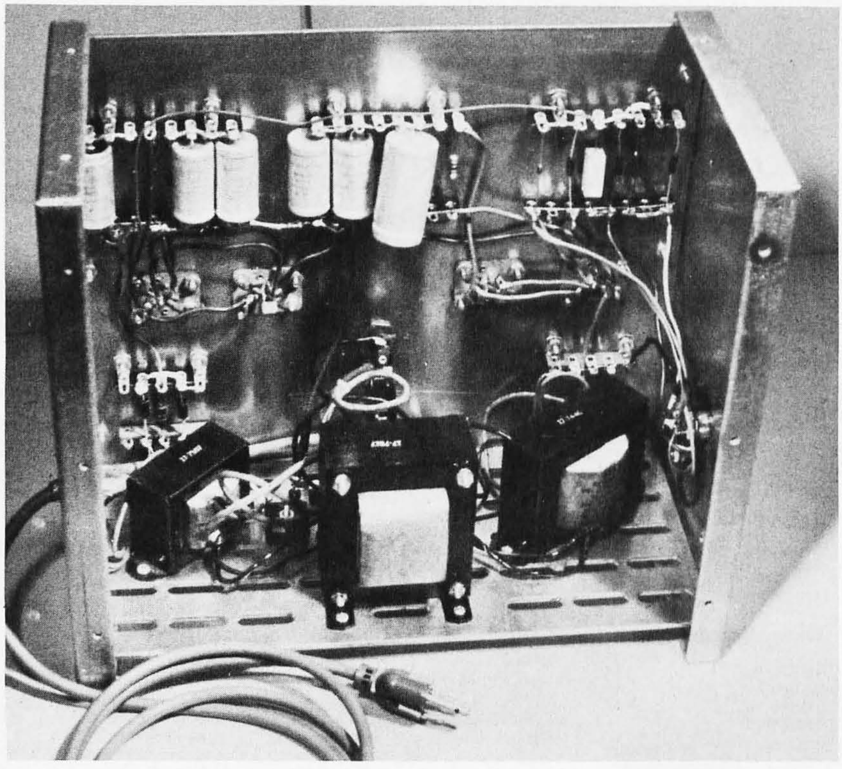


Photo 1: The Sphere system features a completely separate power supply. All regulation is done in the unit shown here, which eliminates the need for on board regulators and the extra heat dissipation on logic cards.

I ended up ordering a "SYS2/KIT – Intelligent" which included a Motorola 6800 processor with 4 KB of programmable memory, full keyboard, TV interface, a cassette interface and what was probably the biggest selling point, 1 KB of EROM which had software for what Sphere calls a "mini assembler," text editor and a debug routine. All this (during the special introductory period) for only $750. Clearly a bargain!

By mid August I had received confirmation of my order and a promised delivery date of the second week in October. Part way through this 60 day wait I heard a rumor that Sphere had not delivered any systems and that all the company consisted of was two people in a garage. When the second week in October arrived and my kit didn't, I put in my first call to Utah and discovered that it would be about two weeks late because they hadn't been able to get all the parts from their suppliers on time to put the kits together. October 30, as promised, brought a box from Sphere which had most of the electronics (all except the cassette interface) but no metalwork; so the next day I again called, this time to find out about the metalwork and two or three parts which were not exactly as specified on the parts list. Fortunately all the parts were usable substitutions and the metalwork arrived the same day so I commenced to build.

Sockets are provided for all the expensive MOS devices, but I decided to use sockets on

everything so for the next four days I soldered in sockets in all my spare time. I was soon ready to plug it in except that the interconnection and power cables had been back ordered and had not come yet. So instead I inspected the boards again, modified a TV for monitor use and waited for a week until they arrived.

Finally I was ready to try out my system; and sweating profusely, I threw the power switch. When no smoke appeared I let out a sigh of relief and then began trying to find out why it wasn't doing what I thought it should be doing.

The television interface seemed to be working, but all that was showing on the screen was a set of random characters and none of the keys would affect anything. Looking at the address lines with a scope, it became obvious that the processor was stuck in a loop. One of the things noticeably lacking from the manual was a set of debugging procedures for the hardware, so again I called Utah. In response to the question of debugging procedures, I was told that it was almost certainly a solder splash on the address lines, although it was possibly the refresh clock notworking.

Even though I had already checked for solder splashes at least twice I went back and looked again just to make sure. When no splashes turned up and the refresh clock appeared to be working correctly, I tried finding out what memory locations were being addressed in the loop (no simple task with only a single trace scope). Finally I pulled out the address drivers (here is where my foresight in using sockets paid off) and manually input the addresses which were in the loop to see what was in these locations in the PROM. Sure enough, with the program which was in the PROM, once the microprocessor got to this point (which it would upon startup), there was no way to get out of the loop.

Armed with what I thought would be convincing data I again called Sphere and this time was asked to send in the PROMs along with a $5 shipping and handling charge. Two weeks later the PROMs came back with the explanation that someone must have forgotten to plug in a master when one of the PROMs had been programmed. I plugged in the chips again, turned on the power supply and lo and behold I had a blinking cursor and an otherwise blank screen. Well, an *almost* blank screen anyway. Several nonblinking cursor blocks would not clear from the screen but would only move around. I decided to ignore these for a while as they seemed to be only an annoyance and nothing disastrous.

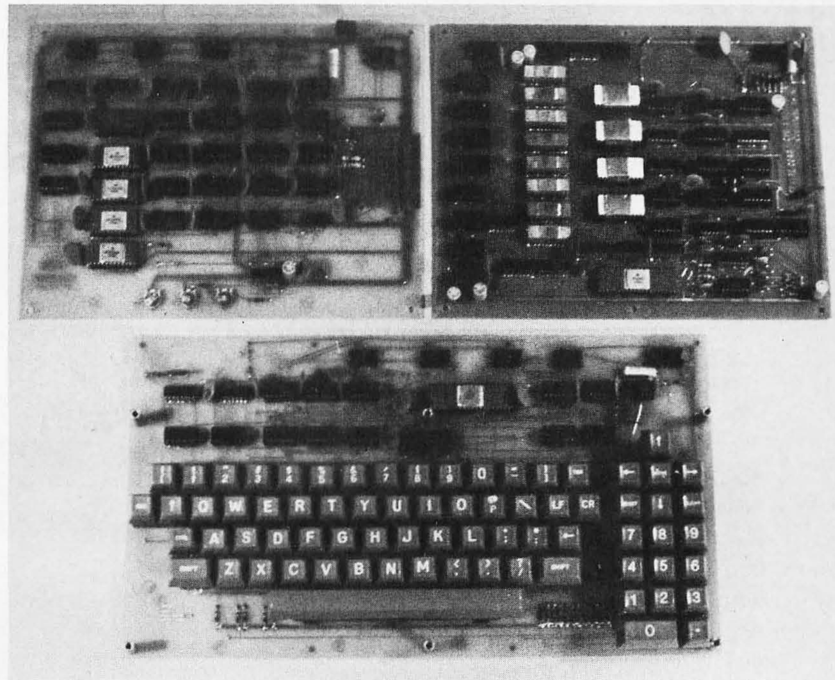


Photo 2: Here are the three main boards of the Sphere system. At the upper left is the video interface board with 512 bytes of memory used for the display. At upper right is the main processor board with the 6800, 4 KB of dynamic memory and 1 KB of EROM. **The** *lower board is the keyboard assembly, which includes a numerical keypad and cursor control keys as well as a full alphanumeric keyboard.*

A short while later I had the editor program working as well as the debug program for opening and changing the contents of memory locations. Unfortunately there was little documentation about the "mini assembler" and I could not figure out how to use this feature based on the original documentation package. So after a few days of inputting simple programs through the debug routine, I went back to trying to fix the TV display.

By this time the problem had become a little more than annoying. On four of the 16 lines of display, several blocks would not clear as noted above, but far worse, if the cursor was placed on any of these lines, more blocks would be created randomly in the two or three positions following the cursor. On four other lines several characters would not be displayed as the correct symbol. By swapping memory ICs, I determined that the problem was in the M6810 chips used for character storage in the TV display. Rather than waiting two weeks for replacements from Sphere, I bought two new ones from an outside vendor. One of these also had a minor defect, so I was running a 50% failure rate on these chips.

I decided to call Sphere and find out about the "mini assembler." It turned out this call was unnecessary since a manual update covering the "mini assembler" and



Photo 3: It works! Here is my completed Sphere system in operation driving a small video monitor (not purchased from Sphere). The board interconnections are made with flat ribbon cables, and the keyboard is shown mounted in its case.

including a memory map had been mailed the day before I called. I also asked about the problems with the character memories and was told that actually the M6810 was not specified to be fast enough for the application but that because Motorola generally made the chips better than spec, they had decided to use them instead of the slightly more expensive M6810-1 which was guaranteed to be fast enough. Personally, I would have preferred to pay the extra cost (< $5 extra for the four chips according to Motorola price lists) and have a display that worked the first time.

Finally the manual addition arrived and I could use the "mini assembler" and could start writing programs seriously. Also with the manual update was a notice about the serial interface board which still had not arrived. Shortly after Sphere started shipments, a conference sponsored by BYTE was held to try to determine some standards for cassette interfaces. Sphere then stopped shipment of all their cassette boards so they could redesign the board so it met the standards which were decided on. This meant another wait of about two months, but the eventual cassette interface would meet the provisional industry standards and even included the option of interfacing to two cassettes which the original design did not offer.

Probably the worst fault I found with my Sphere system is its lack of documentation. With the manual addition received in December, a start was made towards correcting this fault. For example, assembly instructions should include some sort of debugging suggestions. The manual I received provided little if any advice on debugging a nonfunctioning system other than: Look for solder splashes. Solder splashes *do* occur, but other faults (like misprogrammed EROMs) also occur and the method of detecting these faults should not be left entirely up to the customer.

When all things are considered, I think Sphere has done a very good job in the short time the company has been in existence. The hardware supplied in the kits is generally of good quality. In the few cases where the hardware is not up to par, they are working to improve it and usually will retrofit kits already shipped when they do. The problems and delays I faced should no longer face future buyers since the Sphere people are hard at work ironing out bugs and glitches in the product and its documentation.

The programs in the EROM provide an incredible amount of flexibility, especially considering that everything is stuffed into only 1 KB of memory; and certainly the keyboard, TV interface and EROM combination make program entry an order of magnitude simpler than the switch and LED front panel method.

Computer systems can get quite complex. I extend my praise to Sphere for getting all the myriad pieces together in such a short time. If a company which has been in business less than a year has no faults greater than these, it will quite likely become one of the leaders of the industry in a very short period of time. ■

the PLEASER...



...we disigned our computers for people like you who are interested in having one, but up 'til now didn't have $30 or $40 thousand to put into one. It is all together in one package now, thanks to advanced technology and the good ole american spirit of doing it faster, better, and cheaper than the other guy.

Because we are a young company (we just celebrated our first birthday) we have a lot of that "spirit" and some pretty capable men who came up with a computer that others are just now beginning to copy. While copying us is a real compliment, no one has yet equalled or bettered what we have developed.

We have a Central Processing Unit (CPU) that contains the most advanced technology available today. That means to you that the CPU will do what a CPU ought to do. It's reliable, and **THAT'S A PLEASER.**

We have a Serial Interface Module (SIM) that is a very powerful idea in the computer world today. Let me explain. You can have an inexpensive audio cassette player / recorder as a mass-memory storage device.Understand the power of the SIM board. For just pennies you can store vast amounts of Data on an ordinary audio cassette. **NOW THAT'S A PLEASER** Sphere also offers the floppy disc as an option when time gets to be a premium.

We have a Program Development System (PDS) that is an aid to programming. What an idea! It remembers its job even when the computer is off. With instant call-up of the program outline when the power is turned on. There is an Editor, an Assembler, and a Debugger to help you do your programming correctly, and quickly, leaving you a lot more time to actually use the computer as a data handler... and **THAT'S A PLEASER.**

As if that weren't enough ...we have a newly added feature THE 4K ROM BD. A board containing up to 4000 bytes of Read Only Memory. (The Stuff that's on it doesn't go away when the computer is off.) You put your most used applications and computer games on the ROM board Turn on the power, locate your favorite game, and you're up and hummin immediately. This makes using the computer lots of fun for a change. **WHAT A PLEASER.**

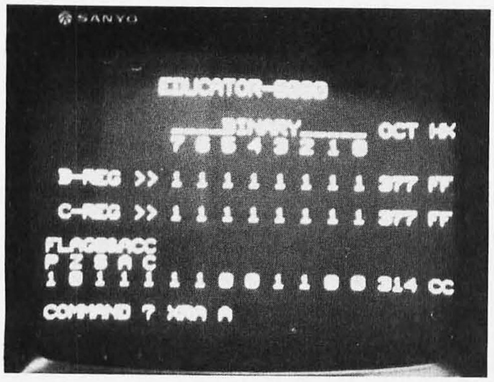
This is what you get. A computer that does what you want it to do when you want it to do it, at a price that you can finally aford. **NOW, THAT'S REALLY A PLEASER!!!**

from SPHERE
the COMPUTER PEOPLE
who have it ALL TOGETHER.

P.S. There's much much more, so write today for prices and information.

SPHERE CORPORATION

940 North 400 East
North Salt Lake,
Utah 84054
(801) 292-8466



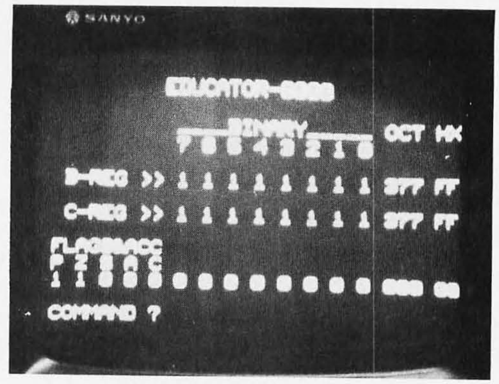


Photo 1: In this example, Educator-8080 is shown before (left) and after (right) the execution of an XRA A instruction. The effect of this instruction is of course to clear the accumulator A, as is shown at the right.

Explore an 8080 with Educator-8080

Charles P Howerton
President, Digital Group Software Systems Inc
PO Box 1086
Arvada CO 80001

What Is an Educator?

Educator-8080 was designed as a classroom instruction aid for a microprocessor programming course. The principal design goals were to develop a system which would illustrate the architecture of the machine and the effect of the execution of various instructions. For example, the reader might ask to what use the logical EXCLUSIVE OR function may be put in an 8080. This function, which operates on each bit, has a value of 1 if either of the two operands or arguments, but not both, has a value of 1; otherwise it has a value of 0. The Educator-8080 can simply illustrate this function. In the example shown in photo 1 (left and right), both arguments of this function are equal: the first argument is the value in the accumulator (A) and the second argument is also the value stored in the accumulator (A). The function value (ie: the result) is placed in the accumulator after execution. The left photo shows the accumulator (and other registers, etc, which are not affected) before execution of the instruction XRA A, and the right photo shows it after execution. The result is that the accumulator (A) is cleared, ie: it contains eight 0 bits. This result is consistent with the definition of the EXCLUSIVE OR above: Whenever both bit arguments are 0, or are 1, a value of 0 is returned. This example shows that the Educator-8080 is a convenient means to illustrate rather complex operations which facilitate learning the instruction set and architecture without the tedium of plowing through books. A subordinate goal was to implement the entire system with the exception of the physical input output routines and the stack in 1024 bytes of memory. All of the design goals were met. In addition, if the IO devices are ASCII oriented, a reduction in the length of the error messages (perhaps limiting them to the error code number) should provide sufficient space for the inclusion of the physical input output routines and the stack within the 1 K byte memory space.

Educator-8080 is written in a fairly straightforward manner and it should not be particularly difficult to adapt it to any 8080 system with more than 1 K bytes of pro-

grammable memory, a keyboard, and an output device of some kind. It is designed to operate with a television display device and to dynamically show the results of the execution of the input commands.

It would probably be desirable to modify the display function somewhat if the output device is a hard copy device such as a Teletype. The content could be the same but the elimination of blank lines and printing the titles only every 10 or 15 instructions would speed things up considerably on a Teletype device. The input output routines required to adapt Educator-8080 to almost any system are described functionally but are not given in detail. They should be adapted from routines already in use for a given system.

The Instruction Set

The Educator-8080 instruction set is a subset of the 8080 instruction set. The commands implemented within Educator-8080 were selected to provide representative instructions from most of the functional instruction groups. Since the instructions are to be executed one at a time from keyboard input, there was no need to incorporate any of the Jump, Call or Return instructions; however, since the flags are displayed after each operation, it is very easy to determine whether or not a given conditional Jump, Call or Return would cause a transfer of control by simply observing the setting for the flag whose status is being tested. In addition, because of memory limitations none of the instructions which cause memory to be read or written were implemented. Finally, no instruction whose action could not be readily observed was implemented.

To keep the display as uncluttered as possible the registers which could be accessed by Educator-8080 instructions were limited to the accumulator and the B and C registers. It would not be particularly difficult to incorporate the rest of the registers into the display and as operands for the Educator-8080 instruction subset. However, unless the ability to address memory is desirable the only instructions which could be added to the subset would be the DAD and the XCHG.

The instruction subset and the valid operands for each instruction are shown in table 1. Table 2 contains the corresponding information as it is loaded into the computer's memory and used by Educator-8080.

Immediate Operands

Almost half of the instructions supported by Educator-8080 require immediate

Table 1: Command List for Educator-8080. In order to illustrate the operations of the 8080 processor, Educator-8080 interprets a subset of the 8080's instructions. The subset generally references the accumulator, A, and registers B or C; it excludes all branching and program control operations. The complete list of available operations is found in this table.

		—Flags—				
Command	**Description of Operation**	**P**	**Z**	**S**	**A**	**C**
ACI i	Add the value of the **Carry Flag** and the value of the immediate operand i to the contents of the **accumulator**.	X	X	X	X	X
ADC r	Add the value of the **Carry Flag** and the contents of register r to the contents of the **accumulator**.	X	X	X	X	X
ADD r	Add the contents of register r to the **accumulator**.	X	X	X	X	X
ADI i	Add the value of the immediate operand i to the **accumulator**.	X	X	X	X	X
ANA r	Logically AND the contents of register r with the **accumulator.**	X	X	X	X	0
ANI i	Logically-AND the value of the immediate operand i with the contents of the **accumulator**.	X	X	X	X	0
CMA	Complement the contents of the **accumulator**, changing all of the zeros to ones and all of the ones to zeros.	N	N	N	N	N
CMC	Complement the value of the **Carry Flag**; if it is zero make it one, or if it is one make it zero.	N	N	N	N	X
CMP r	Compare the contents of register r with the contents of the **accumulator**.	X	X	X	X	X
CPI i	Compare the value of the immediate operand i with the contents of the **accumulator**.	X	X	X	X	X
DAA	Decimal adjust the value in the **accumulator** (after an arithmetic command using decimal numbers).	X	X	X	X	X
DCR r	Decrement (subtract 1 from) the contents of register r	X	X	X	X	N
DCX rp	Decrement the contents of the register pair **rp**.	N	N	N	N	N
INR r	Increment (add 1 to) the contents of register **r**.	X	X	X	X	N
INX rp	Increment the contents of the register pair **rp**.	N	N	N	N	N
MVI r, i	Move the value of the immediate operand i into register **r**.	N	N	N	N	N
MOV r, s	Move the value of the contents of register **s** into register **r** leaving **s** unchanged.	N	N	N	N	N
NOP	No operation: do nothing.	N	N	N	N	N
ORA r	Logically OR the contents of register r with the **accumulator**.	X	X	X	0	0
ORI i	Logically OR the value of the immediate operand i with the **accumulator**.	X	X	X	0	0
RAL	Rotate the contents of the **accumulator** left one bit position with the high order bit going to the **Carry Flag** and the **Carry Flag** going into the low order bit of the **accumulator**.	N	N	N	N	X
RAR	Rotate the contents of the **accumulator** right one bit position with the low order bit of the **accumulator** going into the **Carry Flag** and the **Carry Flag** going into the high order bit of the **accumulator.**	N	N	N	N	X
RLC	Rotate the contents of the **accumulator** left one bit position with the high order bit going into both the low order bit and the **Carry Flag.**	N	N	N	N	X
RRC	Rotate the contents of the **accumulator** right one bit position with the low order bit going into both the high order bit and the **Carry Flag.**	N	N	N	N	X
SBB r	Subtract the values of the **Carry Flag** and register **r** from the **accumulator**.	X	X	X	X	X
SBI i	Subtract the values of the **Carry Flag** and the immediate operand i from the **accumulator.**	X	X	X	X	X
STC	Set the **Carry Flag** to a 1 value.	N	N	N	N	1
SUB r	Subtract the contents of register **r** from the **accumulator**.	X	X	X	X	X
SUI i	Subtract the value of the immediate operand i from **accumulator.**	X	X	X	X	X
XRA r	Logically Exclusive OR the contents of register r with the **accumulator**.	X	X	X	0	0
XRI i	Logically Exclusive OR the value of the immediate operand i with the **accumulator**.	X	X	X	0	0

Key:
- **i** = any valid immediate operand (see text).
- **r** = any one of the three registers displayed A, B, or C.
- **rp** = must be the register pair B and C which is designated B.
- **s** = any one of the three registers displayed A, B, or C

Values for FLAGS:
- X = Changed value depends on operands and command.
- 0 = Reset to zero always.
- 1 = Set to one always.
- N = Not changed by this command.

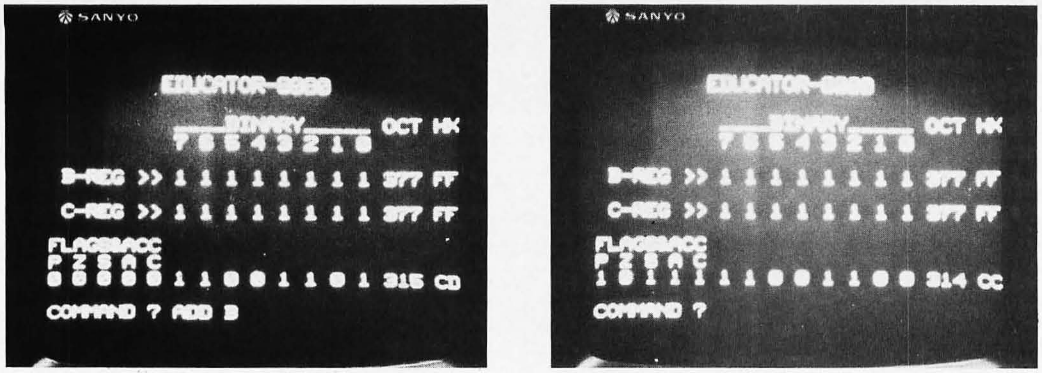


Photo 2: What happens when an 8080 executes an ADD B instruction? A specific example is illustrated in this set of before and after snapshots.

operands. An immediate operand is a constant value which is part of the instruction being executed and it immediately follows the operation code of the instruction, hence the name immediate.

Whenever a single byte "constant" is required in a program, its inclusion as the immediate value of an appropriate instruction reduces the length of the program because there is no need to address the value

Table 2: Operation Code Table for the Educator-8080 program. Table 1 showed the command list for the program. This table gives the absolute machine codes for the command table beginning at address <2>/122. Each command is represented by a 3 byte ASCII character string mnemonic followed by the naked (without register values) 8080 operation code and the address of the routine which interprets the command. The routine name is shown symbolically in the right hand column, and can be found in the program of listing 1.

Address	ASCII Mnemonic	------Octal Code------ Mnemonic	Opcode	Routine	Routine Name
<2>/122	'ACI'	101 103 111	316	212<1>	IMMED
<2>/130	'ADC'	101 104 103	210	152<1>	RG210
<2>/136	'ADD'	101 104 104	200	152<1>	RG210
<2>/144	'ADI'	101 104 111	306	212<1>	IMMED
<2>/152	'ANA'	101 116 101	240	152<1>	RG210
<2>/160	'ANI'	101 116 111	346	212<1>	IMMED
<2>/166	'CMA'	103 115 101	057	144<1>	DIRCT
<2>/174	'CMC'	103 115 103	077	144<1>	DIRCT
<2>/202	'CMP'	103 115 120	270	152<1>	RG210
<2>/210	'CPI'	103 120 111	376	212<1>	IMMED
<2>/216	'DAA'	104 101 101	047	144<1>	DIRCT
<2>/224	'DCR'	104 103 122	005	245<1>	RG543
<2>/232	'DCX'	104 103 130	013	264<1>	RG54B
<2>/240	'INR'	111 116 122	004	245<1>	RG543
<2>/246	'INX'	111 116 130	003	264<1>	RG54B
<2>/254	'MOV'	115 117 126	100	145<1>	MOVRT
<2>/262	'MVI'	115 126 111	006	205<1>	MVIRT
<2>/270	'NOP'	116 117 120	000	144<1>	DIRCT
<2>/276	'ORA'	117 122 101	260	152<1>	RG210
<2>/304	'ORI'	117 122 111	366	212<1>	IMMED
<2>/312	'RAL'	122 101 114	027	144<1>	DIRCT
<2>/320	'RAR'	122 101 122	037	144<1>	DIRCT
<2>/326	'RLC'	122 114 103	007	144<1>	DIRCT
<2>/334	'RRC'	122 122 103	017	144<1>	DIRCT
<2>/342	'SBB'	123 102 102	230	152<1>	RG210
<2>/350	'SBI'	123 102 111	336	212<1>	IMMED
<2>/356	'STC'	123 124 103	067	144<1>	DIRCT
<2>/364	'SUB'	123 125 102	220	152<1>	RG210
<2>/372	'SUI'	123 125 111	326	212<1>	IMMED
<3>/000	'XRA'	130 122 101	250	152<1>	RG210
<3>/006	'XRI'	130 122 111	356	212<1>	IMMED

directly. Immediate values have an implied address which is the address of the byte following the opcode and this address is supplied from the program counter register automatically whenever an immediate type instruction is executed. In the Educator-8080 system the "program counter" is provided by the operator's sequence of commands which are executed one by one.

Educator-8080 has three different types of immediate values as part of the input command and defaults to one of these types if the input command omits type information.

The general form of an immediate operand is as follows:

T p V p

Where: T is the type code which designates the form of the immediate value and may be any of the following:

B – for a binary immediate value

Q – for an octal immediate value

H – for a hexadecimal immediate value

If the type code is omitted entirely and the first nonpunctuation character encountered is a numeric digit 0 to 7, then a default type of octal is assumed.

p is any form of punctuation (eg: single or double quotes, parentheses, etc). Punctuation is not required, and provision for its inclusion is solely in the interest of enabling the user to enter commands in a format consistent with that of various advanced assemblers.

V is the value of the immediate operand expressed in a form consistent with the explicit or implied type selected. The form and content of the value field for each type is as follows:

T = B: V is a series of eight consecutive numeric characters which have the value zero or one.

Example: **B'11000111'.** V is **11000111,** quotes are optional.

T = Q or T omitted: V is a series of

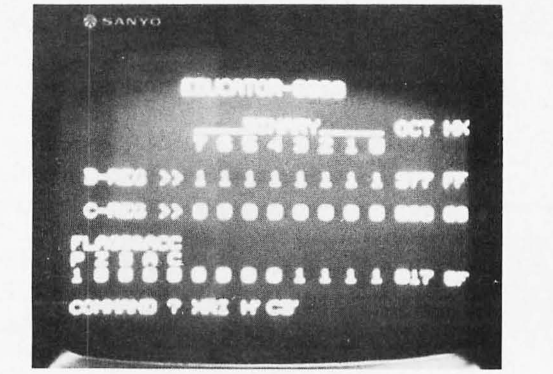




Photo 3: To illustrate the use of hexadecimal immediate values, this photo shows the operation of XRI H'C3'.

Listing 1: The Educator-8080 program expressed as an absolute assembly language listing. The notations <0>, <1>, <2> and <3> are used to denote the high order (page) address bytes of four consecutive pages in memory address space. When loading the program into a given system, these notations become bytes with consecutive octal values. Thus to load the program at location 200/000 in memory address space, the values utilized would be 200, 201, 202 and 203.

```
address    octal-code      label    op     operand       commentary

*The control routine is the top of the structure and controls the operation of the entire program.

<0>/000    061 xxx xxx     CNTRL    LXI    SP, STACK     Set stack pointer to programmable memory;
<0>/003    315 026 <0>     NOTZER   CALL   DSPLY         Display contents of registers;
<0>/006    315 316 <0>              CALL   CMDNT         Enter a command;
<0>/011    315 063 <1>              CALL   FETCH         Fetch the correct opcode;
<0>/014    267                      ORA    A             Set zero flag as per contents;
<0>/015    302 003 <0>              JNZ    NOTZER        Jump if not zero error occurred;
<0>/020    315 030 <2>              CALL   XQTER         Go execute the current command;
<0>/023    303 003 <0>              JMP    NOTZER        Loop forever;

*This display routine controls the generation of the dynamic display.

<0>/026    041 167 <3>     DSPLY    LXI    H,TITLS       Load address of titles into HL;
<0>/031    315 261 <0>              CALL   CHEDT         Display titles;
<0>/034    041 257 <3>              LXI    H,BLINE       Load addr of BLINE title;
<0>/037    072 351 <3>              LDA    BREG          Load contents of BREG into A;
<0>/042    315 132 <0>              CALL   DSPCV         Convert and display;
<0>/045    041 271 <3>              LXI    H,CLINE       Load addr of CLINE title;
<0>/050    072 350 <3>              LDA    CREG          Load contents of CREG into A;
<0>/053    315 132 <0>              CALL   DSPCV         Convert and display;
<0>/056    041 304 <3>              LXI    H,AFHDR       Load addr of A'flags title;
<0>/061    315 261 <0>              CALL   CHEDT         Display titles;
<0>/064    052 346 <3>              LHLD   PSWA          Load flags and A into HL;
<0>/067    175                      MOV    A,L           Move flags to A;
<0>/070    346 004                  ANI    B'00000100'   AND off all but parity flag;
<0>/072    315 237 <0>              CALL   DSPFG         Display the flag value;
<0>/075    175                      MOV    A,L           Move flags to A;
<0>/076    346 100                  ANI    B'01000000'   AND off all but zero flag;
<0>/100    315 237 <0>              CALL   DSPFG         Display the flag value;
<0>/103    175                      MOV    A,L           Move flags to A;
<0>/104    346 200                  ANI    B'10000000'   AND off all but sign flag;
<0>/106    315 237 <0>              CALL   DSPFG         Display the flag value;
<0>/111    175                      MOV    A,L           Move flags to A;
<0>/112    346 020                  ANI    B'00010000'   AND off all but auxiliary carry flag;
<0>/114    315 237 <0>              CALL   DSPFG         Display the flag value;
<0>/117    175                      MOV    A,L           Move flags to A;
<0>/120    346 001                  ANI    B'00000001'   AND off all but carry flag;
<0>/122    315 237 <0>              CALL   DSPFG         Display the flag value;
<0>/125    174                      MOV    A,H           Move A register value to A;
<0>/126    315 137 <0>              CALL   DSPCN         Display with no title print;
<0>/131    311                      RET                  Return to the CNTRL routine;

*The display conversion routine prints binary, octal and hexadecimal.

<0>/132    365             DSPCV    PUSH   PSW           Save output value for CHEDT;
<0>/133    315 261 <0>              CALL   CHEDT         Display line title addr in HL;
<0>/136    361                      POP    PSW           Retrieve saved output value;
<0>/137    036 010         DSPCN    MVI    E,Q'010'      Move 8 to E register;
<0>/141    007             DSPBT    RLC                  Rotate MSB into Carry and LSB;
<0>/142    365                      PUSH   PSW           Save current value;
<0>/143    346 001                  ANI    Q'001'        AND off all but LSB;
<0>/145    315 237 <0>              CALL   DSPFG         Go display bit value;
<0>/150    361                      POP    PSW           Retrieve saved current value;
<0>/151    035                      DCR    E             Decrement loop count;
<0>/152    302 141 <0>              JNZ    DSPBT         Jump if loop count not zero;
<0>/155    267                      ORA    A             Reset carry;
<0>/156    036 003                  MVI    E,Q'003'      Move 3 to E register;
<0>/160    027             DSPQT    RAL                  MSB to Carry, Carry to LSB,
<0>/161    027                      RAL                    do it again,
<0>/162    027                      RAL                      three times for octal digit shift;
<0>/163    365                      PUSH   PSW           Save current value;
<0>/164    346 007                  ANI    Q'007'        AND off all but octal LSD;
<0>/166    366 060                  ORI    Q'060'        OR on bits to make ASCII numeric character;
<0>/170    315 xxx xxx              CALL   CHRPR         Output the character;
<0>/173    361                      POP    PSW           Retrieve saved current value;
<0>/174    035                      DCR    E             Decrement loop count;
<0>/175    302 160 <0>              JNZ    DSPQT         Jump if loop count not zero;
<0>/200    315 251 <0>              CALL   DSPSP         Output a space;
<0>/203    036 002                  MVI    E,Q'002'      Move 2 to E;
<0>/205    007             DSPHT    RLC                  Rotate MSB into Carry and LSB,
<0>/206    007                      RLC                    do it again,
<0>/207    007                      RLC                      four times for,
<0>/210    007                      RLC                        hexadecimal shift;
<0>/211    365                      PUSH   PSW           Save current value;
<0>/212    346 017                  ANI    B'00001111'   AND off all but hexadecimal LSD;
<0>/214    306 060                  ADI    Q'060'        Add on bits to make ASCII numeric character;
<0>/216    376 072                  CPI    Q'072'        Compare result to one more than 9;
<0>/220    332 225 <0>              JC     DSPHS         If numeric then skip adjustment;
<0>/223    306 007                  ADI    Q'007'        Add 7 giving ASCII 'A' thru 'F' codes;
```

three consecutive numeric characters which have octal digit values of from 0 to 7.

Example: **Q'307'** V is **307**, quotes are optional.

T=H: V is a pair of consecutive characters which have hexadecimal digit values from 0 to F.

Example: **H'C7'** V is **C7**, quotes are optional.

With the exception of the move immediate (MVI) command which requires a destination register, immediate commands are entered as the mnemonic opcode followed by the immediate operand in any of its valid forms.

Some "before and after" examples of Educator-8080 commands are shown in photos 1 through 3. In each case, a command is typed into the keyboard of the computer, then the Educator-8080 display following the command is depicted.

Entering Commands

Commands are entered into Educator-8080 as a string of characters (eg: letters, numbers, spaces and punctuation) followed by a command termination character. As written, Educator-8080 assumes that the command termination character will be an ASCII carriage-return (octal 015). However, any other keyboard character code may be used as the command termination character by changing the value of the immediate operand in the instruction located at address <0>/341 which tests for command termination. (See listing 1.)

Since it is not uncommon to make errors when keying information into a computer, two provisions have been made in Educator-8080 for correcting or eliminating

Listing 1, continued:

```
     address    octal-code       label    op     operand     commentary

→   <0>/225   315  xxx  xxx    DSPHS    CALL   CHRPR       Output the character;
    <0>/230   361                       POP    PSW         Retrieve saved current value;
    <0>/231   000                       NOP
    <0>/232   035                       DCR    E           Decrement loop count;
    <0>/233   302  205  <0>             JNZ    DSPHT       Jump if loop count not zero;
    <0>/236   311                       RET                Return to calling routine;

    •Display flag or binary digit followed by a space. Alternate entry is used to display a space.

    <0>/237   312  244  <0>    DSPFG    JZ     DSPFZ       Jump if passed value is a zero;
    <0>/242   076  001                  MVI    A,Q'001'    Otherwise move a 1 into A;
    <0>/244   306  060         DSPFZ    ADI    Q'060'      Convert into ASCII numeric character;
→   <0>/246   315  xxx  xxx             CALL   CHRPR       Output the character;
    <0>/251   365              DSPSP    PUSH   PSW         Save the flags and value in A;
    <0>/252   076  040                  MVI    A,Q'040'    Move space into A;
→   <0>/254   315  xxx  xxx             CALL   CHRPR       Output the space;
    <0>/257   361                       POP    PSW         Retrieve the saved flags and A;
    <0>/260   311                       RET                Return to the calling routine;

    •The character string output edit routine.

    <0>/261   176              CHEDT    MOV    A,M         Move next character into A;
    <0>/262   376  200                  CPI    Q'200'      Compare it to 200 octal;
    <0>/264   310                       RZ                 Return if equal it's end of string;
    <0>/265   322  277  <0>             JNC    CHSPA       Jump if greater for space routine;
→   <0>/270   315  xxx  xxx             CALL   CHRPR       Else go output the character;
    <0>/273   043              CHEND    INX    H           Increment the string index;
    <0>/274   303  261  <0>             JMP    CHEDT       Loop for next character;
    <0>/277   326  200         CHSPA    SUI    Q'200'      Subtract 200 octal from value;
    <0>/301   107                       MOV    B,A         Move space count to B;
    <0>/302   076  040         CHSPL    MVI    A,Q'040'    Move space to A;
→   <0>/304   315  xxx  xxx             CALL   CHRPR       Output the space;
    <0>/307   005                       DCR    B           Decrement space count;
    <0>/310   302  302  <0>             JNZ    CHSPL       Jump if count not zero to start of loop;
    <0>/313   303  273  <0>             JMP    CHEND       Jump back into CHEDT loop;

    •The command entry routine accepts input from the keyboard for commands.

    <0>/316   041  332  <3>    CMDNT    LXI    H,CMDMS     Move address of 'COMMAND ? ' to HL;
    <0>/321   315  261  <0>             CALL   CHEDT       Display the message;
    <0>/324   041  352  <3>             LXI    H,CMDAR     Move address of command input area HL;
    <0>/327   006  026                  MVI    B,Q'026'    Move maximum length to B;
→   <0>/331   315  xxx  xxx    CMDKB    CALL   KEYBD       Get an input character;
    <0>/334   376  014                  CPI    Q'014'      Is it a control-1 line delete?
    <0>/336   312  000  <0>             JZ     CNTRL       If so then restart program;
    <0>/341   376  015                  CPI    Q'015'      Is it a carriage return?
    <0>/343   312  376  <0>             JZ     CMDND       If so then go compress input;
    <0>/346   376  177                  CPI    Q'177'      Is it a delete character?
    <0>/350   302  355  <0>             JNZ    CMDST       If not then go store the character;
    <0>/353   076  033                  MVI    A,Q'033'    If so replace with back arrow;
    <0>/355   167              CMDST    MOV    M,A         Store input character in command buffer;
→   <0>/356   315  xxx  xxx             CALL   CHRPR       Display the input character;
    <0>/361   043                       INX    H           Increment command work area index;
    <0>/362   005                       DCR    B           Decrement command length count;
    <0>/363   302  331  <0>             JNZ    CMDKB       If not full then reiterate;
    <0>/366   076  001                  MVI    A,Q'001'    If buffer full then select error
    <0>/370   315  063  <2>             CALL   ERROR         number 1 and print its message;
    <0>/373   303  000  <0>             JMP    CNTRL       Restart the program;

    •The command compress routine eliminates all but letters and numbers.

    <0>/376   041  352  <3>    CMDND    LXI    H,CMDAR     Load HL with address of work area;
    <1>/001   345                       PUSH   H           Push & pop move it to DE
    <1>/002   321                       POP    D             as the compression pointer;
    <1>/003   076  026                  MVI    A,Q'026'    Load A with maximum length;
    <1>/005   220                       SUB    B           Subtract remaining length from B;
    <1>/006   107                       MOV    B,A         Move actual length to B;
    <1>/007   176              CMDNX    MOV    A,M         Move command character to A;
    <1>/010   376  033                  CPI    Q'033'      Is it a back arrow (character delete)?
    <1>/012   302  027  <1>             JNZ    CMDCH       If not then go to other tests;
    <1>/015   076  352                  MVI    A,CMDAR-L   Low address byte of CMDAR to A;
    <1>/017   273                       CMP    E           Compare to current low address byte;
    <1>/020   322  055  <1>             JNC    CMDNS       If not greater then skip save;
    <1>/023   033                       DCX    D           Else back up compression pointer;
    <1>/024   303  055  <1>             JMP    CMDNS       Skip saving the character;
    <1>/027   376  060         CMDCH    CPI    Q'060'      Is the character less than '0'?
    <1>/031   332  055  <1>             JC     CMDNS       If so then skip saving it;
    <1>/034   376  072                  CPI    Q'072'      Is the character less than '9' + 1?
    <1>/036   332  053  <1>             JC     CMDSV       If so then save numeric value;
    <1>/041   376  101                  CPI    Q'101'      Is the character less than 'A'?
    <1>/043   332  055  <1>             JC     CMDNS       If so then skip saving it;
    <1>/046   376  133                  CPI    Q'133'      Is the character greater than 'Z'?
    <1>/050   322  055  <1>             JNC    CMDNS       If so then skip saving it;
    <1>/053   022              CMDSV    STAX   D           Store character in compressed area;
    <1>/054   023                       INX    D           Increment compression pointer index;
    <1>/055   043              CMDNS    INX    H           Increment input string pointer;
    <1>/056   005                       DCR    B           Decrement actual length count;
    <1>/057   302  007  <1>             JNZ    CMDNX       If length is not zero then reiterate;
    <1>/062   311                       RET                Else return to CNTRL calling point;

    •The FETCH instruction/command routine validates and builds the object code.

    <1>/063   041  122  <2>    FETCH    LXI    H,OPTAB     Load address of opcode table HL;
    <1>/066   036  037                  MVI    E,Q'037'    Move table element count to E;
    <1>/070   345              FLOOP    PUSH   H           Save current element address;
    <1>/071   001  352  <3>             LXI    B,CMDAR     Load address of CMDAR into BC;
    <1>/074   026  003                  MVI    D,Q'003'    Move opcode length to D;
    <1>/076   012              FCOMP    LDAX   B           Load command character to A indexed by B;
    <1>/077   276                       CMP    M           Compare it to table character;
    <1>/100   302  125  <1>             JNZ    FNXEL       If not equal then go to next element;
    <1>/103   003                       INX    B           Increment command character index;
    <1>/104   043                       INX    H           Increment table character index;
    <1>/105   025                       DCR    D           Decrement opcode length counter;
    <1>/106   302  076  <1>             JNZ    FCOMP       If not zero continue test loop;
    <1>/111   343                       XTHL               Exchange HL with top of stack;
    <1>/112   341                       POP    H           Pop HL from stack to clear it;
    <1>/113   136                       MOV    E,M         Move naked opcode to E, D is zero;
    <1>/114   325                       PUSH   D           Save naked opcode;
    <1>/115   043                       INX    H           Increment table pointer;
    <1>/116   136                       MOV    E,M         Decode routine low address byte to E;
    <1>/117   043                       INX    H           Increment table pointer;
    <1>/120   126                       MOV    D,M         Decode routine high address byte to D;
    <1>/121   353                       XCHG               Move decode routine address to HL;
```

errors. The ASCII delete character code (octal 177) is used to delete the last remaining character in the input string. Since a deleted character is not considered to exist, N consecutive delete characters will delete the N preceding characters. For example, if the delete character is shown as a back arrow (←), RAX←L will be reduced to RAL and CQP←←MA will be reduced to CMA. Characters which have been keyed in are displayed after they have been tested. The display function uses the octal value 177 as a clear screen control code; therefore, character deletes are transformed into the back arrow before they are displayed and stored. Educator-8080 users with systems which have a back arrow (octal 033) key on their keyboards may use it as a character delete code and it will have the same effect as the delete key assumed in this version. Users who have neither of these keys can designate any keyboard character as the delete character code by changing the immediate operand in the instruction located at <0>/346 which tests for the delete character. (See listing 1). The other, and somewhat more drastic, method of eliminating keying errors is to delete the entire input line. This is usually done when an error is detected before the command termination character is input but several characters after the error occurred. The procedure for deleting an entire line is to enter an ASCII form feed code (octal 014) which is a "control L" combination on typical ASCII keyboards. This will clear the input line and restart the command entry procedure. Like the command termination and the character delete codes, the line delete code can be made to be any keyboard character by changing the value of the immediate operand in the instruction at location <0>/334 which tests for the line delete code.

A very useful feature of Educator-8080 permits the user to execute the last command input several times. This is accomplished by simply keying the command termination character when the system calls for the entry of a new command. In order to provide this facility the input buffer is not cleared prior to calling for the entry of a new command, so the last previously entered command is still in the buffer. This feature is especially handy when demonstrating the effect of multiple executions of the rotate, increment, decrement, arithmetic and logical commands.

The general format for entering a command is as follows:

OPCODE[p OPERAND-1[p OPERAND-2]]t

Where:

OPCODE   is the mnemonic opcode for the

command. For example; MOV XRI, etc.

p — is any desired form of punctuation or a space. p is not required and, therefore, may be omitted entirely.

OPERAND-1 — is the first or only operand required by an instruction. It may be a register identification or an immediate value. See table 1 for the operand requirements.

OPERAND-2 — is the second operand where required by a specific instruction. See table 1.

t — is the command termination character, an ASCII carriage return in the listing 1 version of Educator-8080.

The brackets ([ ]) shown in the general format are used to indicate that the items within them are optional, since some commands do not require any operands (eg: RAL, STC, CMA, etc), some require one operand only (eg: ADI, CMP, XRA, etc), and some commands require two operands (eg: MOV and MVI).

Error Messages

In the process of entering and executing commands under Educator-8080 there are a number of errors which can occur. When this happens an error message is displayed on the output device. For the benefit of users with television displays, a delay of approximately two seconds occurs as the message is being displayed, to provide time to read it. After the two second delay the normal Educator-8080 display is generated and the command entry mode is reentered. Teletype or other hard copy users will probably wish to alter the error display routine slightly by eliminating the extraneous spaces which are used to center the error messages on the TV monitor screen.

The errors which can occur are listed in absolute octal form in table 3. The error numbers and extended explanations of conditions are as follows:

1. **INPUT TOO LONG:** The input string exceeds 22 characters in length probably because too many characters were deleted since delete character codes count as input characters. Twenty two characters should be sufficient for any normal entry including punctuation and several character deletes.
2. **INVALID COMMAND:** The input command mnemonic is not one of the ones implemented by Educator-8080.
3. **INVALID REGISTER:** The operand register is not A, B or C for a command which requires a single register as an operand or it was not B

Listing 1, continued:

```
address    octal-code     label   op    operand      commentary

<1>/122    321                    POP   D            Unsave naked opcode to DE;
<1>/123    257                    XRA   A            Clear A, no error code;
<1>/124    351                    PCHL               Jump to address of decode routine;
<1>/125    001 006 000    FNXEL   LXI   B,Q'000006'  Load double length 6 into BC;
<1>/130    341                    POP   H            Unsave current element address;
<1>/131    011                    DAD   B            Add 6 to it;
<1>/132    035                    DCR   E            Decrement table element count;
<1>/133    302 070 <1>            JNZ   FLOOP        Reiterate to test next element;
<1>/136    076 002                MVI   A,Q'002'     Move error code 2 to A;
<1>/140    303 063 <2>            JMP   ERROR        Go display error 2, opcode unknown;
<1>/143    000                    NOP                No operation filler;

*The instruction decoder routines follow.

*Instructions using the DIRCT routine require no decoding. Example RAL, CMA, etc.

<1>/144    311            DIRCT   RET                Return to CNTRL for execution;

*The MOVRT is used only by the MOV command.

<1>/145    315 245 <1>    MOVRT   CALL  RG543        Validate destination register;
<1>/150    267                    ORA   A            Set flags based on A contents;
<1>/151    300                    RNZ                Return not zero with error;
                                                     Else fall thru to RG210;

*Instructions using the RG210 routine require a source register.

<1>/152    012            RG210   LDAX  B            Load next command character into A;
<1>/153    003                    INX   B            Increment command character index;
<1>/154    315 173 <1>            CALL  REGAN        Analyze for valid register;
<1>/157    322 166 <1>            JNC   RGERR        If CY=0 then register not valid;
<1>/162    203                    ADD   E            Add naked opcode to register value;
<1>/163    137                    MOV   E,A          Move result back to E;
<1>/164    257                    XRA   A            Clear A indicating no errors;
<1>/165    311                    RET                Return to CNTRL;

*The register error routine is used to indicate register designation errors.

<1>/166    076 003        RGERR   MVI   A,Q'003'     Move error code 3 to A;
<1>/170    303 063 <2>            JMP   ERROR        Go display error 3, invalid register;

*The register analysis and validation routine is used by RG543, RG210 and RG54B.

<1>/173    326 101        REGAN   SUI   Q'101'       Subtract an 'A' from the character;
<1>/175    376 003                CPI   Q'003'       Compare the result to 3;
<1>/177    320                    RNC                If not less than 3 return with CY=0;
<1>/200    075                    DCR   A            Decrement result: A=377, B=000, C=001;
<1>/201    346 007                ANI   Q'007'       AND off all but octal LSD;
<1>/203    067                    STC                Set CY=1 indicating no error;
<1>/204    311                    RET                Return to calling routine;

*The MVIRT is used only by the MVI command.

<1>/205    315 245 <1>    MVIRT   CALL  RG543        Validate destination register;
<1>/210    267                    ORA   A            Set flags based on A contents;
<1>/211    300                    RNZ                Return not zero with error;
                                                     Else fall thru to IMMED;

*Instructions requiring an immediate operand use the IMMED routine.

<1>/212    012            IMMED   LDAX  B            Load next command character into A;
<1>/213    003                    INX   B            Increment command character index;
<1>/214    376 102                CPI   Q'102'       Is the command character a 'B'?
<1>/216    312 301 <1>            JZ    BINRY        If so then process as binary;
<1>/221    376 121                CPI   Q'121'       Is the command character a 'Q'?
<1>/223    312 336 <1>            JZ    OCTAL        If so then process as octal;
<1>/226    376 110                CPI   Q'110'       Is the command character an 'H'?
<1>/230    312 367 <1>            JZ    HEX          If so then process as hexadecimal;
<1>/233    376 070                CPI   Q'070'       Is the command character less than '8'?
<1>/235    332 335 <1>            JC    OCTAD        If so then treat as octal;
<1>/240    076 005                MVI   A,Q'005'     Move error code 5 to A;
<1>/242    303 063 <2>            JMP   ERROR        Go display error 5, invalid immediate;

*Instructions using the RG543 routine require a destination register.

<1>/245    012            RG543   LDAX  B            Load next command character into A;
<1>/246    003                    INX   B            Increment command character index;
<1>/247    315 173 <1>            CALL  REGAN        Analyze for valid register;
<1>/252    322 166 <1>            JNC   RGERR        If CY=0 then register not valid;
<1>/255    007                    RLC                Shift octal register value
<1>/256    007                    RLC                  left three
<1>/257    007                    RLC                    places;
<1>/260    203                    ADD   E            Add naked opcode to shifted value;
<1>/261    137                    MOV   E,A          Move result back to E;
<1>/262    257                    XRA   A            Clear A indicating no errors;
<1>/263    311                    RET                Return to calling routine;

*Instructions using the RG54B routine are INX and DCX.

<1>/264    012            RG54B   LDAX  B            Load next command character into A;
<1>/265    003                    INX   B            Increment command character index;
<1>/266    315 173 <1>            CALL  REGAN        Analyze for valid register;
<1>/271    376 000                CPI   Q'000'       Is the register a zero?
<1>/273    310                    RZ                 If so it's 'B' so return;
<1>/274    076 004                MVI   A,Q'004'     Move error code 4 to A;
<1>/276    303 063 <2>            JMP   ERROR        Go display error 4, invalid register;

*The BINRY routine converts a binary immediate value into usable form.

<1>/301    046 010        BINRY   MVI   H,Q'010'     Move 8 to H for count;
<1>/303    012            BLOOP   LDAX  B            Load next command character into A;
<1>/304    326 060                SUI   Q'060'       Subtract a '0' from it;
<1>/306    376 002                CPI   Q'002'       Is the result less than 2?
<1>/310    322 330 <1>            JNC   IMMER        If not then go display immediate error;
<1>/313    345                    PUSH  H            Save the count;
<1>/314    152                    MOV   L,D          Move D to L (immediate byte);
<1>/315    051                    DAD   H            Shift HL left one bit;
<1>/316    205                    ADD   L            Add L to bit in A;
```

Listing 1, continued:

```
address     octal-code      label     op     operand          commentary

<1>/317     127                       MOV    D,A              Move the result back to D;
<1>/320     341                       POP    H                Unsave the count;
<1>/321     003                       INX    B                Increment command character index;
<1>/322     045                       DCR    H                Decrement the count;
<1>/323     302 303 <1>               JNZ    BLOOP            If not zero then reiterate;
<1>/326     257                       XRA    A                Clear A indicating no errors;
<1>/327     311                       RET                     Return to CNTRL;

*The immediate error routine is used to indicate immediate value errors.

<1>/330     076 006         IMMER     MVI    A,Q'006'         Move error code 3 to A;
<1>/332     303 063 <2>               JMP    ERROR            Go display error 3, invalid immediate;

*The OCTAD entry point to the OCTAL routine is for the default condition.

<1>/335     013             OCTAD     DCX    B                Decrement command character index;

*The OCTAL routine converts an octal immediate value into usable form.

<1>/336     046 003         OCTAL     MVI    H,Q'003'         Move a 3 into H for count;
<1>/340     012             OLOOP     LDAX   B                Load next command character into A;
<1>/341     326 060                   SUI    Q'060'           Subtract a '0' from it;
<1>/343     376 010                   CPI    Q'010'           Is command character less than 8?
<1>/345     322 330 <1>               JNC    IMMER            If not then go display immediate error;
<1>/350     345                       PUSH   H                Save the count;
<1>/351     152                       MOV    L,D              Move D to L immediate byte;
<1>/352     051                       DAD    H                Shift immediate
<1>/353     051                       DAD    H                  byte left
<1>/354     051                       DAD    H                    three bits;
<1>/355     205                       ADD    L                Add L to value in A;
<1>/356     127                       MOV    D,A              Move result back to D;
<1>/357     341                       POP    H                Unsave the count;
<1>/360     003                       INX    B                Increment command character index;
<1>/361     045                       DCR    H                Decrement the count;
<1>/362     302 340 <1>               JNZ    OLOOP            If not zero then reiterate;
<1>/365     257                       XRA    A                Clear A indicating no errors;
<1>/366     311                       RET                     Return to CNTRL;

*The HEX routine converts a hexadecimal immediate value into usable form.

<1>/367     046 002         HEX       MVI    H,Q'002'         Move a 2 into H for count;
<1>/371     012             HLOOP     LDAX   B                Load next command character into A;
<1>/372     326 060                   SUI    Q'060'           Subtract a '0' from it;
<1>/374     376 012                   CPI    Q'012'           Is it less than '9' + 1?
<1>/376     332 010 <2>               JC     HCHOK            If so then numeric character is OK;
<2>/001     326 007                   SUI    Q'007'           Else convert alphabetic to numeric;
<2>/003     376 020                   CPI    Q'020'           Is character value greater than 15?
<2>/005     322 330 <1>               JNC    IMMER            If so then invalid hexadecimal value;
<2>/010     345             HCHOK     PUSH   H                Save the count;
<2>/011     152                       MOV    L,D              Move D to L immediate byte;
<2>/012     051                       DAD    H                Shift immediate
<2>/013     051                       DAD    H                  byte left
<2>/014     051                       DAD    H                    four
<2>/015     051                       DAD    H                      bits;
<2>/016     205                       ADD    L                Add L to value in A;
<2>/017     127                       MOV    D,A              Move result back to D;
<2>/020     341                       POP    H                Unsave the count;
<2>/021     003                       INX    B                Increment command character index;
<2>/022     045                       DCR    H                Decrement the count;
<2>/023     302 371 <1>               JNZ    HLOOP            If not zero then reiterate;
<2>/026     257                       XRA    A                Clear A indicating no errors;
<2>/027     311                       RET                     Return to CNTRL;

*The XQTER routine executes the generated object code for Educator-8080.

<2>/030     353             XQTER     XCHG                    Move generated opcode to HL;
<2>/031     042 046 <2>               SHLD   XQTOP            Store it at execution point;
<2>/034     052 346 <3>               LHLD   PSWA             Load working PSW & A into HL;
<2>/037     345                       PUSH   H                Push & pop sets values for
<2>/040     361                       POP    PSW                working register and flags;
<2>/041     052 350 <3>               LHLD   BANDC            Load working B and C into HL;
<2>/044     345                       PUSH   H                Push & pop sets values for
<2>/045     301                       POP    B                  working B and C registers;
<2>/046     000             XQTOP     NOP                     The command to be executed;
<2>/047     000                       NOP                     Immediate value or NOP;
<2>/050     305                       PUSH   B                Push B and C working register values;
<2>/051     341                       POP    H                Pop them into HL;
<2>/052     042 350 <3>               SHLD   BANDC            Store them in save area;
<2>/055     365                       PUSH   PSW              Push PSW and A working values;
<2>/056     341                       POP    H                Pop them into HL;
<2>/057     042 346 <3>               SHLD   PSWA             Store them in save area;
<2>/062     311                       RET                     Return to CNTRL for next command;

*The ERROR routine is used to display error messages.

<2>/063     365             ERROR     PUSH   PSW              Save error code in A;
<2>/064     041 162 <3>               LXI    H,ERRSP          Load address of error header spaces;
<2>/067     315 261 <0>               CALL   CHEDT            Go output error header spaces
<2>/072     361                       POP    PSW              Unsave error code;
<2>/073     041 014 <3>               LXI    H,ERTAB          Load address of error message table;
<2>/076     205                       ADD    L                Add low address byte to error code;
<2>/077     157                       MOV    L,A              Move result to L, points to offset;
<2>/100     156                       MOV    L,M              Move offset to L;

*Note: HL now contains the address of the error message.

<2>/101     315 261 <0>               CALL   CHEDT            Output the error message;
<2>/104     021 000 000     ERTIM     LXI    D,Q'000000'      Load DE with timing loop value;
<2>/107     035                       DCR    E                Decrement value in E 256 times;
<2>/110     302 105 <2>               JNZ    ERTIM+1          Reiterate loop 256 times;

*The above JMP goes to the first 000 in the LXI command which is an effective NOP.

<2>/113     025                       DCR    D                Decrement D;
<2>/114     302 105 <2>               JNZ    ERTIM+1          Reiterate outer loop 256 times;
<2>/117     076 377                   MVI    A,Q'377'         Move a 377 to A indicating error;
<2>/121     311                       RET                     Return to CNTRL;

*Note: for Teletype or hard copy output bytes <2>/104 thru <2>/116 can be replaced by 000 NOPs.
```

for the INX or DCX commands which require register pairs as operands.

4. **INVALID IMMED TYPE:** The type code for an immediate operand is not B, Q, or H, or if the default was attempted the first digit of the implied octal value was not a digit from 0 to 7.
5. **INVALID IMMED VALUE:** One of the characters in the immediate operand value string was inconsistent with the immediate type code. For example, a digit in a binary input string was not a zero or a one. This can also be caused by not providing the correct quantity of digits for the immediate type specified; too few digits will possibly cause a problem. If too many digits are entered only the first N will be used (N=8 for binary, N=3 for octal and N=2 for hexadecimal).
6. **ERROR!** This message should not occur unless a grave internal error occurs in Educator-8080.

Educator-8080 Program Listing

The Educator-8080 program is presented in an assembly language format as listing 1. It was hand assembled and, therefore, some liberties were taken in the way it was presented. Addresses are shown in a split octal ("Intelese") format of page and address within page. Educator-8080 requires four contiguous 256 byte pages of memory (it just fits); to ease the implementation process all addresses and address sensitive bytes are shown with relative page numbers in the format <P>, where P is a 0, 1, 2 or 3. A simple process of substitution as the program is being put into the machine will provide the ability to locate Educator-8080 in any four contiguous pages provided the program begins on a page boundary.

The assignment of three addresses is left to the user. These three addresses are shown symbolically in both the source and the object code. The first address is for the location of the STACK; insert the address of the stack in the command at location <0>/000. The stack should be capable of being at least 10 to 12 levels deep to function correctly. The second and third addresses are the addresses of the physical input and output routines which must be provided by the user. These routine addresses are shown symbolically as KEYBD and CHRPR in the source listing. The values are shown as 'XXX XXX' in the object code.

Input and Output Routines

The Educator-8080 program references two subroutines for the purpose of exe-

Table 3:

Error Messages. This table consists of a list of address offsets (location <3>/014) followed by the ASCII error message strings. The octal values 201 through 377 are used to encode from 1 to 177 spaces (1 to 127 decimal). The strings contain a single space for these codes. The octal value 200 is used to indicate end of string, and is shown symbolically as the character "▽". The octal value 177 is used to indicate the clear screen operation, and is shown symbolically as the character "■".

Address	Octal Code	ASCII String Value
<3>/014	153 024 043 063 063 104 127 153	Address Offsets for messages 0 through 7
<3>/024	111 116 120 125 124 040 124 117	'INPUT TOO LONG ▽'
<3>/034	117 040 114 117 116 107 200	
<3>/043	111 116 126 101 114 111 104 040	'INVALID COMMAND▽ '
<3>/053	103 117 115 115 101 116 104 200	
<3>/063	111 116 126 101 114 111 104 040	'INVALID REGISTER ▽ '
<3>/073	122 105 107 111 123 124 105 122	
<3>/103	200	
<3>/104	111 116 126 101 114 111 104 040	'INVALID IMMED TYPE ▽ '
<3>/114	111 115 115 105 104 040 124 131	
<3>/124	120 105 200	
<3>/127	111 116 126 101 114 111 104 040	'INVALID IMMED VALUE ▽ '
<3>/137	111 115 115 105 104 040 126 101	
<3>/147	114 125 105 200	
<3>/153	105 122 122 117 122 041 200	'ERROR! ▽ '

The following string is given the name "ERRSP" and is used to clear the screen, then space down to the center prior to displaying an error message.

<3>/162	177 377 377 211 200	'■ ▽ '

cuting IO operations. The KEYBD subroutine is used to read a single character of input from an ASCII keyboard device. The CHRPR subroutine is used to display (or print) a single character. These routines are not shown in the listings, but should be adapted from the routines normally used with the particular system in which the program is run. Both KEYBD and CHRPR use the accumulator (A) to pass a single character argument. KEYBD defines a value in A obtained from the input device. CHRPR displays the value in A on a device such as a video display or Teletype. All other registers of the 8080 processor should be left unchanged upon return from either of these routines. Entry to the IO routines is shown using a CALL instruction in these listings. A corresponding RET instruction in the routine should return control when either operation is completed. An alternate method of entry would be to employ the 8080 RST instruction in place of CALL. If the Educator-8080 listings accompanying this article are used without reassembly, then the CALL instructions would be replaced by an RST and two single byte NOP instructions.

The keyboard entry routine KEYBD

Continued on page 75

Address	Octal Code	Name	ASCII Value
<3>/167	177 211 105 104 125 103 101 124	TITLS	'■ EDUCAT '
<3>/177	120 122 055 070 060 070 060 264		'OR-8080 '
<3>/207	137 137 137 137 102 111 116 101		'←←←← BINA'
<3>/217	122 131 137 137 137 137 137 040		'RY←←←←← '
<3>/227	117 103 124 040 110 130 212 067		'OCT HX 7'
<3>/237	040 066 040 065 040 064 040 063		' 6 5 4 3 '
<3>/247	040 062 040 061 040 060 250 200		' 2 1 0 ▽ '
<3>/257	102 055 122 105 107 040 076 076	BLINE	'B-REG >> '
<3>/267	040 200		' ▽ '
<3>/271	241 103 055 122 105 107 040 076	CLINE	' C-REG '
<3>/301	076 040 200		'> ▽ '
<3>/304	240 106 114 101 107 123 046 101	AFHDR	' FLAGS&A '
<3>/314	103 103 227 120 040 132 040 123		'CC P Z S'
<3>/324	040 101 040 103 227 200		' A C ▽ '
<3>/332	240 103 117 115 115 101 116 104	CMDMS	'COMMAND '
<3>/342	040 077 040 200		'? ▽ '

Table 4: Educator-8080 standard display format messages. This table contains the definitions of several character string messages which are used to format the output display device. As in table 3, the codes from octal 201 to 377 represent from 1 to 177 spaces transmitted. The character "▽" is used to indicate an end of text code, octal 200. The character "■" is used to indicate a clear screen code, octal 177.

MACHINE LANGUAGE

Chapter I

THE '8008' CPU INSTRUCTION SET

The '8008' microprocessor has quite a comprehensive instruction set that consists of 48 basic instructions, which, when the possible permutations are considered, result in a total set of about 170 instructions.

The instruction set allows the user to direct the computer to perform operations with memory, with the seven basic registers in the CPU, and with INPUT and OUTPUT ports.

It should be pointed out that the seven basic registers in the CPU consist of one "accumulator," a register that can perform mathematical and logic operations, plus an additional six registers, which, while not having the full capability of the accumulator, can perform various useful operations. These operations include the ability to hold data, serve as an "operator" with the accumulator, and increment or decrement their contents. Two of these six registers have special significance because they may be used to serve as a "pointer" to locations in memory.

The seven CPU registers have arbitrarily been given symbols so that we may refer to them in an abbreviated language. The first register is designated by the symbol 'A' in the following discussion and will be referred to as the "accumulator" register. The next four registers will be referred to as the 'B,' 'C,' 'D' and 'E' registers. The remaining two special memory pointing registers shall be designated the 'H' (for the HIGH portion of a memory address) and the 'L' (for the LOW portion of a memory address) registers.

The CPU also has several "flip-flops" which shall be referred to as "FLAGS." The flip-flops are set as the result of certain operations and are important because they can be "tested" by many of the instructions with the instruction's meaning changing as a consequence of the particular status of a FLAG at the time the instruction is executed. There are four basic flags which will be referred to in this manual. They are defined as follows:

The 'C' flag refers to the carry bit status. The carry bit is a one unit register which changes state when the accumulator overflows or underflows. This bit can also be set to a known condition by certain types of instructions. This is important to remember when developing a program because quite often a program will have a long string of instructions which do not utilize the carry bit or care about its status, but which will be causing the carry bit to change its state from time-to-time. Thus, when one prepares to do a series of operations that will rely on the carry bit, one often desires to set the carry bit to a known state.

The 'Z' for zero flag refers to a one unit register that when desired will indicate whether the value of the accumulator is exactly equal to zero. In addition, immediately after an increment of decrement of the B, C, D, E, H or L registers, this flag will also indicate whether the increment or decrement caused that particular register to go to zero.

The 'S' for sign flag refers to a one unit register that indicates whether the value in the accumulator is a positive or negative value (based on two's complement nomenclature). Essentially, this flag monitors the most significant bit in the accumulator and is "set" when it is a one.

The 'P' flag refers to the last flag in the group which is for indicating when the accumulator contains a value which has even parity. Parity is useful for a number of reasons and is usually used in conjunction with testing for error conditions on words of data especially when transferring data to and from external devices. Even parity occurs when the number of bits that are a logic one in the accumulator is an even value. Zero is considered an even value for this purpose. Since there are eight bits in the accumulator, even parity will occur when zero, two, four or six bits are in the logic one condition regardless of what order they may appear in within the register.

It is important to note that the Z, S, and P flags (as well as the previously mentioned C flag) can all be set to known states by certain instructions. It is also important to note that some instructions do not result in the flags being set so that if the programmer desires to have the program make decisions based on the status of flags, the programmer should ensure that the proper instruction, or sequence of instructions is utilized. It is particularly important to note that load register instructions do not by themselves set the flags. Since it is often desirable to obtain a data word (that is, load it into the accumulator) and test its status for such parameters as whether or not the value is zero, or a negative number, and so forth, the programmer must remember to follow a load instruction by a logical instruction (such as the NDA - "and the accumulator") in order to set the flags before using an instruction that is conditional in regards to a flag's status.

The description of the various types of instructions available using an '8008' CPU which follows will provide both the machine language code for the instruction given as three octal digits, and also a mnemonic name suitable for writing programs in "symbolic" type language which is usually easier than trying to remember octal codes! It may be noted that the symbolic language used is the same as that originally suggested by Intel Corporation which developed the '8008' CPU-on-a-chip. Hence users who may already be familiar with the suggested mnemonics will not have any relearning problems and those learning the mnemonics for the first time will have plenty of good company. If the programmer is not already aware of it, the use of mnemonics facilitates working with an "assembler" program when it is desired to develop relatively large and complex programs. Thus the programmer is urged to concentrate on learning the mnemonics for the instructions and not waste time memorizing the octal codes. After a program has been written using the mnemonic codes, the programmer can always use a lookup table to convert to the machine code if an assembler program is not available. It's a lot easier technique (and less subject to error) than trying to memorize

PROGRAMMING FOR THE "8008"

and similar microcomputers

the 170 or so three digit combinations which make up the machine instruction code set!

The programmer must also be aware, that in this machine, some instructions require more than one word in memory. "Immediate" type commands require two consecutive words. JUMP and CALL commands require three consecutive words. The remaining types only require one word.

The first group of instructions to be presented are those that are used to load data from one CPU register to another, or from a CPU register to a word in memory, or vice-versa. This group of instructions requires just one word of memory. It is important to note that none of the instructions in this group affect the flags.

LOAD DATA FROM ONE CPU REGISTER TO ANOTHER CPU REGISTER

MNEMONIC	MACHINE CODE
LAA	300
LBA	310
.	.
.	.
LAB	301

The load register group of instructions allows the programmer to move the contents of one CPU register into another CPU register. The contents of the originating (from) register is not changed. The contents of the destination (to) register becomes the same as the originating register. Any CPU register can be loaded into any CPU register. Note that loading register A into register A is essentially a NOP (no operation) command. When using mnemonics the load symbol is the letter L followed by the "to" register and then the "from" register. The mnemonic LBA means that the contents of register A (the accumulator) is to be loaded into register B. The mnemonic LAB states that register B is to have its contents loaded into register A. It may be observed that this basic instruction has many variations. The machine language coding for this instruction is in the same format as the mnemonic code except that the letters used to represent the registers are replaced by numbers that the computer can use. Using octal code, the seven CPU registers are coded as follows:

Register A = 0
Register B = 1
Register C = 2
Register D = 3
Register E = 4
Register H = 5
Register L = 6

Also, since the machine can only utilize numbers, the octal number '3' in the most significant location of a word signifies that the computer is to perform a "load" operation. Thus, in machine coding, the instruction for loading register B with the contents of register A becomes '310' (in octal form). Or, if one wanted to get very detailed, the actual binary coding for the eight bits of information in the instruction word would be '11 001 000.' It is important to note that the load instructions do not affect any of the flags.

LOAD DATA FROM ANY CPU REGISTER TO A LOCATION IN MEMORY

LMA	370
LMB	371
LMC	372
LMD	373
LME	374
LMH	375
LML	376

This instruction is very similar to the previous group of instructions except that now the contents of a CPU register will be loaded into a specified memory location. The memory location that will receive the contents of the particular CPU register is that whose address is specified by the contents of the CPU H and L registers at the time the instruction is executed. The H CPU register specifies the HIGH portion of the address desired, and the L CPU register specifies the LOW portion of the address into which data from the selected CPU register is to be loaded. Note that there are seven different instructions in this group. Any CPU register can have its contents loaded into any location in memory. This group of instructions does not affect any of the flags.

LOAD DATA FROM A MEMORY LOCATION TO ANY CPU REGISTER

LAM	307
LBM	317
LCM	327
LDM	337
LEM	347
LHM	357
LLM	367

This group of instructions can be considered the opposite of the previous group. Now, the contents of the word in memory whose address is specified by the H (for HIGH portion of the address) and L (LOW portion of the address) registers will be loaded into the CPU register specified by the instruction. Once again, this group of instructions has no affect on the status of the flags.

LOAD IMMEDIATE DATA INTO A CPU REGISTER

LAI	006
LBI	016
LCI	026
LDI	036
LEI	046
LHI	056
LLI	066

An IMMEDIATE type of instruction requires two words in order to be completely specified. The first word is the instruction itself. The second word, or "immediately following" word, must contain the data upon which "immediate" action is taken. Thus, a load IMMEDIATE instruction in this group means that the contents of the word immediately following the instruction word is to be loaded into the specified register. For example, a typical load immediate instruction would be LAI 001. This would result in the value 001 (octal) being placed in the A register when the instruction was executed. It is important to remember that all IMMEDIATE type in-

structions MUST be followed by a data word. An instruction such as LDI by itself would result in improper operation because the computer would assume the next word contained data. If the programmer had mistakenly left out the data word, and in its place had another instruction, the computer would not realize the operator's mistake. Hence the program would be fouled-up! Note too, that the load immediate group of instructions does not affect the flags.

LOAD IMMEDIATE DATA INTO A MEMORY LOCATION

LMI	076

This instruction is essentially the same as the load immediate into the CPU register group except that now, using the contents of the H and L registers as "pointers" to the desired address in memory, the contents of the "immediately following word" will be placed in the memory location specified. This instruction does not affect the status of the flags.

The above rather large group of LOAD instructions permits the programmer to direct the computer to move data about. They are used to bring in data from memory where it can be operated on by the CPU. Or, to temporarily store intermediate results in the CPU registers during complicated and extended calculations, and of course allow data, such as results, to be placed back into memory for long term storage. Since none of them will alter the contents of the four CPU flags, these instructions can be called upon to set up data before instructions that may affect or utilize the flag's status are executed. The programmer will use instructions from this set frequently. The mnemonic names for the instructions are easy to remember as they are well ordered. The most important item to remember about the mnemonics is that the TO register is always indicated first in the mnemonic, and then the FROM register. Thus LBA equals "load TO register B FROM register A.

INCREMENT THE VALUE OF A CPU REGISTER BY ONE

INB	010
INC	020
IND	030
INE	040
INH	050
INL	060

This group of instructions allows the programmer to add one to the present value of any of the CPU registers except the accumulator. (Note carefully that the accumulator can NOT be incremented by this type of instruction. In order to add one to the accumulator a mathematical addition instruction, described later, must be used.) This instruction for incrementing the defined CPU registers is very valuable in a number of applications. For one thing, it is an easy way to have the L register successively "point" to a string of locations in memory. A feature that makes this type of instruction even more powerful is that the result of the incremented register will affect the Z, S, and P flags. (It will not change the C or "carry"flag.) Thus, after a CPU register has been incremented by this instruction, one can utilize a flag test instruction (such as the conditional JUMP and CALL instructions to be described later) to determine whether that particular register has a value of zero (Z flag), or if it is a negative number (S flag), or even parity (P flag). It is important to note that this group of instructions, and the decrement group (described in the next paragraph) are the only instructions which allow the flags to be manipulated by operations that are not concerned with the accumulator (A) register.

DECREMENT THE VALUE OF A CPU REGISTER BY ONE

DCB	011
DCC	021
DCD	031
DCE	041
DCH	051
DCL	061

The DECREMENT group of instructions is similar to the INCREMENT group except that now the value one will be subtracted from the specified CPU register. This instruction will not affect the C flag. But, it does affect the Z, S, and P flags. It should also be noted that this group, as with the increment group, does not include the accumulator register. A separate mathematical instruction must be used to subtract one from the accumulator.

ARITHMETIC INSTRUCTIONS USING THE ACCUMULATOR

The following group of instructions allow the programmer to direct the computer to perform arithmetic operations between other CPU registers and the accumulator, or between the contents of words in memory and the accumulator. All of the operations for the described addition, subtraction, and compare instructions affect the status of the flags.

ADD THE CONTENTS OF A CPU REGISTER TO THE ACCUMULATOR

ADA	200
ADB	201
ADC	202
ADD	203
ADE	204
ADH	205
ADL	206

This group of instructions will simply ADD the present contents of the accumulator register to the present value of the specified CPU register and leave the result in the accumulator. The value of the specified register is unchanged except in the case of the ADA instruction. Note that the ADA instruction essentially allows the programmer to double the value of the accumulator (which is the A register)! If the addition causes an overflow or underflow then the carry (C flag) will be affected.

ADD THE CONTENTS OF A CPU REGISTER PLUS THE VALUE OF THE CARRY FLAG TO THE ACCUMULATOR

ACA	210
ACB	211
ACC	212
ACD	213
ACE	214
ACH	215
ACL	216

This group is identical to the previous group except that the content of the carry flag is considered as an additional bit (MSB) in the specified CPU register. The combined value of the carry bit plus the contents of the specified CPU register are added to the value in the accumulator. The results are left in the accumulator. Again, with the exception of the ACA instruction, the contents of the specified CPU register are left unchanged. Again too, the carry bit (C flag) will be affected by the results of the operation.

SUBTRACT THE CONTENTS OF A CPU REGISTER FROM THE ACCUMULATOR

SUA	220
SUB	221
SUC	222
SUD	223
SUE	224
SUH	225
SUL	226

This group of instructions will cause the present value of the specified CPU register to be subtracted from the value in the accumulator. The value of the specified register is not changed except in the case of the SUA instruction. (Note that the SUA instruction is a convenient instruction with which to "clear" the accumulator.) The carry flag will be affected by the results of a SUBTRACT instruction.

SUBTRACT THE CONTENTS OF A CPU REGISTER AND THE VALUE OF THE CARRY FLAG FROM THE ACCUMULATOR

SBA	230
SBB	231
SBC	232
SBD	233
SBE	234
SBH	235
SBL	236

This group is identical to the previous group except that the content of the carry flag is considered as an additional bit (MSB) in the specified CPU register. The combined value of the carry bit plus the contents of the specified CPU register are SUBTRACTED from the value in the accumulator. The results are left in the accumulator. The carry

bit (C flag) is affected by the result of the operation. With the exception of the SBA instruction the content of the specified CPU register is left unchanged.

COMPARE THE VALUE IN THE ACCUMULATOR AGAINST THE CONTENTS OF A CPU REGISTER

CPA	270
CPB	271
CPC	272
CPD	273
CPE	274
CPH	275
CPL	276

The COMPARE group of instructions are a very powerful and somewhat unique set of instructions. They direct the computer to compare the contents of the accumulator against another register and to set the flags as a result of the comparing operation. It is essentially a subtraction operation with the value of the specified register being subtracted from the value of the accumulator except that the value of the accumulator is not actually altered by the operation. However, the flags are set in the same manner as though an actual subtraction operation had occured. Thus, by subsequently testing the status of the various flags after a COMPARE instruction has been executed, the program can determine whether the compare operation resulted in a match or non-match. In the case of a non-match, one may determine if the compared register contained a value greater or less than that in the accumulator. This would be accomplished by testing the Z flag and C flag respectively utilizing a conditional JUMP or CALL instruction (which will be described later).

ADDITION, SUBTRACTION, AND COMPARE INSTRUCTIONS THAT USE WORDS IN MEMORY AS OPERANDS

The five types of mathematical operations: ADD, ADD with CARRY, SUBTRACT, SUBTRACT with CARRY, and COMPARE, which have just been presented for the cases where they operate with the contents of CPU registers, can all be performed with words that are in memory. As with the LOAD instructions that operate with memory, the H and L registers must contain the address of the word in memory that it is desired to ADD, SUBTRACT, or COMPARE to the accumulator. The same conditions for the operations as was detailed when using the CPU registers apply. Thus, for mathematical operations with a word in memory, the following instructions are used.

ADD THE CONTENTS OF A MEMORY WORD TO THE ACCUMULATOR

ADM 207

ADD THE CONTENTS OF A MEMORY WORD PLUS THE VALUE OF THE CARRY FLAG TO THE ACCUMULATOR

ACM 217

SUBTRACT THE CONTENTS OF A MEMORY WORD FROM THE ACCUMULATOR

SUM 227

SUBTRACT THE CONTENTS OF A MEMORY WORD AND THE VALUE OF THE CARRY FLAG FROM THE ACCUMULATOR

SBM 237

COMPARE THE VALUE IN THE ACCUMULATOR AGAINST THE CONTENTS OF A MEMORY WORD

CPM 277

IMMEDIATE TYPE ADDITIONS, SUBTRACTIONS, AND COMPARE INSTRUCTIONS

The five types of mathematical operations discussed above can also be performed with the operand being the word of data immediately after the instruction. This group of instructions is similar in format to the previously described LOAD IMMEDIATE instructions. The same conditions for the mathematical operations as discussed for the operations with the CPU registers apply.

ADD IMMEDIATE

ADI 004

ADD WITH CARRY IMMEDIATE

ACI 014

SUBTRACT IMMEDIATE

SUI 024

SUBTRACT WITH CARRY IMMEDIATE

SBI 034

COMPARE IMMEDIATE

CPI 074

LOGICAL INSTRUCTIONS WITH THE ACCUMULATOR

There are several groups of instructions which allow BOOLEAN LOGIC operations to be performed between the contents of the CPU registers and the A (accumulator) register. In addition there are logic IMMEDIATE type instructions. The boolean logic operations are valuable in a number of programming applications. The instruction set allows three basic boolean operations to be performed. These are: the LOGICAL AND, the LOGICAL OR, and the EXCLUSIVE OR operations. Each type of logic operation is performed on a bit-by-bit basis between the accumulator and the CPU register or memory location specified by the instruction. A detailed explanation of each type of logic operation, and the appropriate instructions for each type is presented below. The logic instruction set is also valuable because all of them will cause the C (carry) flag to be placed in the zero condition. This is important if one is going to perform a sequence of instructions that will eventually use the status of the C flag to arrive at a decision as it allows the programmer to set the C flag to a known state at the start of the sequence. All other flags are set in accordance with the result of the logic operation. Hence, the group often has value when the programmer desires to determine the contents of a register that has just been loaded into a register. (Since the load instructions do not alter the flags.)

THE BOOLEAN 'AND' OPERATION INSTRUCTION SET

When the boolean AND instruction is executed, each bit of the accumulator will be compared with the corresponding bit in he register or memory location specified by the instruction. As each bit is compared a logic result will be placed in the accumulator for each bit comparison. The logic result is determined as follows. If both the bit in the accumulator and the bit in the register with which the operation is being performed are a logic one, then the accumulator bit will be left in the logic one condition. For all other possible combinations (A bit equals one, X bit equals zero; A bit equals zero, X bit equals one; or A bit equals zero, X bit equals zero), then the accumulator bit will be cleared to the zero state. An example will illustrate the logical AND operation.

INITIAL STATE OF THE ACCUMULATOR

10 101 010

CONTENTS OF OPERAND REGISTER

11 001 100

FINAL STATE OF THE ACCUMULATOR

10 001 000

There are seven logical AND instructions that allow any CPU register to be used as the AND operand. They are as follows.

NDA	240
NDB	241
NDC	242
NDD	243
NDE	244
NDH	245
NDL	246

The contents of the operand register is not altered by an AND logical instruction.

There is also a logical AND instruction that allows a word in memory to be used as an operand. The address of the word in memory that will be used is pointed to by the contents of the H and L CPU registers.

NDM	247

And finally there is also a logical AND IMMEDIATE type of instruction that will use the contents of the word immediately following the instruction as the operand.

NDI	044

The next group of boolean logic instructions direct the computer to perform the logical OR operation on a bit-by-bit basis with the accumulator and the contents of a CPU register or a word in memory. The logical OR operation will result in the accumulator having a bit set to a logic one if either that bit in the accumulator, or the corresponding bit in the operand register is a logic one. Since the case where both the accumulator bit and operand bit are a one also satisfies the criteria, that condition will also result in the accumulator bit being left in the one state. If neither register has a logic one in the bit position, then the accumulator bit for that position remains in the zero state. An example illustrates the results of a logical OR operation.

INITIAL STATE OF THE ACCUMULATOR

10 101 010

CONTENT OF THE OPERAND REGISTER

11 001 100

FINAL STATE OF THE ACCUMULATOR

11 101 110

There are seven logical OR instructions that allow any CPU register to be used as the OR operand.

ORA	260
ORB	261
ORC	262
ORD	263
ORE	264
ORH	265
ORL	266

By using the H and L registers as pointers one can also use a word in memory as an OR operand.

ORM	267

There is also the logical OR IMMEDIATE instruction.

ORI	064

As with the logical AND group of instructions, the logical OR instruction does not alter the contents of the operand register.

The last group of boolean logic instructions is a variation of the logic OR. The variation is termed the EXCLUSIVE OR logical operation. The EXCLUSIVE OR operation is similar to the OR except that when the corresponding bits in both the accumulator and the operand register are a one then the accumulator bit will be cleared to zero. Thus, the accumulator bit will be a one after the operation only if just one of the registers (accumulator register or operand register) has a one in the bit position. (Again, the operation is performed on a bit-by-bit basis.) An example provides clarification.

INITIAL STATE OF THE ACCUMULATOR

10 101 010

CONTENTS OF THE OPERAND REGISTER

11 001 100

FINAL STATE OF THE ACCUMULATOR

01 100 110

The seven instructions that allow the CPU registers to be used as operands are shown next.

XRA	250
XRB	251
XRC	252
XRD	253
XRE	254
XRH	255
XRL	256

The instruction that uses registers H and L as pointers to a memory location is:

XRM	257

And the EXCLUSIVE OR IMMEDIATE type instruction is:

XRI	054

As in the case of the logical OR operation, the operand register is not altered except for the special case when the XRA instruction is used. This instruction, which directs the computer to EXCLUSIVE OR the accumulator with itself, will cause the operand register, since it is the accumulator, to have its contents altered (unless it should happen to be zero at the time the instruction is executed). This is because, regardless of what value is in the accumulator, if it is EXCLUSIVE OR'ed with itself, the result will be zero! The example below illustrates the specific operation.

ORIGINAL VALUE OF ACCUMULATOR

10 101 010

EXCLUSIVE OR'ed WITH ITSELF

10 101 010

FINAL VALUE OF ACCUMULATOR

00 000 000

This only occurs when the logical EXCLUSIVE OR is performed on the accumulator itself. It can be shown that the results of performing the logical OR or logical AND between the accumulator and itself will result in the original accumulator value being retained.

INSTRUCTIONS FOR ROTATING THE CONTENTS OF THE ACCUMULATOR

It is often desirable to be able to shift the contents of the accumulator either right or left. In a fixed length register, a simple shift operation would result in some information being lost because what was in the MSB or LSB (depending on in which direction the shift occured) would be shifted right out of the register! Therefore, instead of just shifting the contents of a register, an operation termed ROTATING is utilized. Now, instead of just shifting a bit off the end of the register, the bit is brought around to the other end of the register. For instance, if the register is rotated to the right, the LSB (least significant bit) would be brought around to the position of the MSB (most significant bit) which would have been vacated by the shifting of its original contents to the right. Or, in the case of a shift to the left, the MSB would be brought around to the position of the LSB.

The carry bit (C flag) can be considered as an extension of the accumulator register. The instruction set for this machine allows two types of ROTATE instructions. One considers the carry bit to be part of the accumulator register for the rotate operation. The other type does not. In addition, each type of rotate can be done either to the right or to the left.

It should be noted that the rotate operations are particularly valuable when it is desired to multiply a number or divide a number. This is because shifting the contents of a register to the left effectively multiplies a binary number by a power of two. Shifting a binary number to the right provides the inverse operation.

ROTATING THE ACCUMULATOR LEFT

RLC	002

Rotating the accumulator left with the RLC instruction means the MSB of the accumulator will be brought around to the LSB position and all other bits will be shifted one position to the left. While this instruction does not shift through the carry bit, the carry bit will be set by the status of the MSB of the accumulator at the start of the ROTATE LEFT operation. (This feature allows the programmer to determine what the MSB was prior to the shifting operation by testing the C flag after the rotate instruction has been executed.

ROTATING THE ACCUMULATOR LEFT THROUGH THE CARRY BIT

RAL 022

The RAL instruction will cause the MSB of the accumulator to go into the carry bit. The initial value of the carry bit will be shifted around to the LSB of the accumulator. All other bits are shifted one position to the left.

ROTATING THE ACCUMULATOR RIGHT

RRC 012

The RRC instruction is similar to the RLC instruction except that now the LSB of the accumulator is placed in the MSB of the accumulator. All other bits are shifted one position to the right. Also, the carry bit will be set to the initial value of the LSB of the accumulator at the start of the operation.

ROTATING THE ACCUMULATOR RIGHT THROUGH THE CARRY BIT

RAR 032

Here, the LSB of the accumulator is brought around to the carry bit. The initial value of the carry bit is shifted to the MSB of the accumulator. All other bits are shifted a position to the right.

It should be noted that the C flag is the only flag that is altered by a rotate instruction. All other flags remain unchanged.

JUMP INSTRUCTIONS

The instructions discussed so far have all been "direct action" instructions. The programmer arranges a sequence of these types of instructions in memory. When the program is started the computer proceeds to execute the instructions in the order in which they are encountered. The computer automatically reads the contents of a memory location, executes the instruction it finds there, and then automatically increments a special address register called a PROGRAM COUNTER that will result in the machine reading the information contained in the next sequential memory location. However, it is often desirable to perform a series of instructions located in one section of memory, and then skip over a group of memory locations and start executing instructions in another section of memory. This action can be accomplished by a group of instructions that will cause a new address value to be placed in the PROGRAM COUNTER. This will cause the computer to go to a new section of memory and then execute instructions sequentially from the new memory location.

The JUMP instructions in this computer add considerable power to the machine's capabilities because there are a series of "conditional" JUMP instructions available. That is, the computer can be directed to test the status of a particular FLAG (C, Z, S or P). If the status of the flag is the desired one, then a JUMP will be performed. If it is not, the machine will continue to execute the next instruction in the current sequence. This capability provides a means for the computer to make "decisions" and to modify its operation as a function of the status of the various flags at the time that a program is being executed.

In a manner similar to IMMEDIATE types of instructions, the JUMP instructions require more than one word of memory. A JUMP instruction requires three words to be properly defined. (Remember that IMMEDIATE type instructions required two words.) The JUMP instruction itself is the first word. The second word must contain the LOW ADDRESS portion of the address of the word in memory that the PROGRAM COUNTER is to be set to point to, which is the new location from which the next instruction is to be fetched. The third word must contain the HIGH ADDRESS (sometimes referred to as the PAGE) of the memory address that the program counter will be set to. That is, the high order portion of the address in memory that the computer will JUMP to in order to obtain its next instruction.

THE UNCONDITIONAL JUMP INSTRUCTION

JMP 1X4

Note: The machine code 1X4 indicates that any code for the second octal digit of the machine code is valid. It is recommended as a standard practice that the code '0' be used. Thus, the typical machine code would be 104.

Remember, the JUMP instruction must be followed by two more words which contain the LOW, and then the HIGH (PAGE) portion of the address that the program is to JUMP to!

JUMP IF THE DESIGNATED FLAG IS TRUE (CONDITIONAL JUMP)

JTC	140
JTZ	150
JTS	160
JTP	170

As with the UNCONDITIONAL JUMP instruction, the CONDITIONAL JUMP instructions must be followed by two words of information. The LOW portion, then the HIGH portion, of the address that program execution is to continue from if the jump is executed. The JUMP IF TRUE group of instructions will only jump to the designated address if the condition of the appropriate flag is TRUE (logical one). Thus, the JTC instruction states that if the carry flag (C) is a logical one (TRUE) then the jump is to be executed. If it is a logical zero (FALSE) then program execution is to continue with the next instruction in the current sequence of instructions. In a similar manner the JTZ instruction states that if the ZERO FLAG is TRUE then the jump is to be performed. Otherwise the next instruction in the present sequence is executed. Likewise for the JTS and JTP instructions.

JUMP IF THE DESIGNATED FLAG IS FALSE (CONDITIONAL JUMP)

JFC	100
JFZ	110
JFS	120
JFP	130

As with all JUMP instructions these instructions must be followed by the LOW address then the HIGH address of the memory location that program execution is to continue from if the jump is executed. This group of instructions is the opposite of the jump if the flag is true group. For instance, the JFC instruction commands the computer to test the status of the carry (C) flag. If the flag is FALSE (a logic zero), then the jump is to be performed. If it is TRUE, then program execution is to continue with the next instruction in the current sequence of instructions. The same procedure holds for the JFZ, JFS and JFP instructions.

SUBROUTINE CALLING INSTRUCTIONS

Quite often when a programmer is developing computer programs the programmer will find that a particular algorithm (sequence of instructions for performing a function) can be used many times in different parts of the program. Rather than having to keep entering the same sequence of instructions at different locations in memory, which would not only consume the time of the programmer, but would also result in a lot of memory being used to perform one particular function, it is desirable to be able to be able to put an often used sequence of commands in just one location in memory. Then, whenever the particular algorithm is required by another part of the program, it would be convenient to jump to the section that contained the often used algorithm, perform the sequence of instructions, and then return back to the main part of the program. This is a standard practice in computer operations. A frequently used algorithm can be designated a SUBROUTINE. A special set of instructions allows the programmer to CALL a SUBROUTINE. In other words, specify a special type of JUMP command that will eventually allow the program to RETURN to the original "jumping" point in the program. A second type of instruction is used to terminate a SUBROUTINE. This special terminator will cause the program to revert back and pick up the next sequential in-

struction in memory that immediately follows the original CALLING instruction. A great deal of computer power is provided by the instruction set in this machine that allows one to CALL and RETURN from SUBROUTINES. This is because, in a manner similar to that provided for the CONDITIONAL JUMP instructions, there are a number of CONDITIONAL CALL and CONDITIONAL RETURN commands in the instruction set.

Like the JUMP instructions, the CALL instructions all require three words in order to be fully specified. The first word is the CALL instruction itself. The next two words must contain the LOW and HIGH portions of the starting address of the subroutine that is being "called."

When a CALL instruction is encountered by the computer, the CPU will actually save the current value of the PROGRAM COUNTER by storing it in a special PROGRAM COUNTER PUSH-DOWN STACK. This stack is capable of holding six addresses plus the current operating address. What this means is that the machine is capable of "nesting" up to seven subroutines at one time. Thus, one can have a subroutine, that in turn calls another subroutine, that in turn calls another one, up to seven levels, and the machine will still be able to return to the initial calling location. The programmer must ensure that subroutines are not nested more than seven levels otherwise the PROGRAM COUNTER PUSH-DOWN STACK will push the original calling address(es) completely out of the push-down stack. The program could then no longer automatically return to the initial calling location.

The RETURN instruction which terminates a SUBROUTINE only requires one word. When the CPU encounters a RETURN instruction it causes the PROGRAM COUNTER PUSH-DOWN STACK to "pop" up one level. This effectively causes the address saved in the stack by the calling routine to be taken as the new program counter. Hence, program execution returns to the calling location.

THE UNCONDITIONAL CALL INSTRUCTION

CAL	1X6

This instruction followed by two words containing the LOW and then the HIGH order of the starting address of the SUBROUTINE that is to be executed is an UNCONDITIONAL CALL. The subroutine will be executed regardless of the status of the FLAGS. The next sequential address after the CAL instruction is saved in the PROGRAM COUNTER PUSH-DOWN STACK.

THE UNCONDITIONAL RETURN INSTRUCTION

RET	0X7

This instruction directs the CPU to unconditionally "pop" the program counter push-down stack UP one level. Program execution will continue from the address saved by the subroutine calling instruction.

CALL A SUBROUTINE IF THE DESIGNATED FLAG IS TRUE

CTC	142
CTZ	152
CTS	162
CTP	172

In a manner similar to the conditional JUMP IF TRUE instructions, these instructions (which must all be followed by the LOW and HIGH portions of the called subroutine's starting address) will only perform the "call" if the designated flag is in the TRUE (logical one) state. If the designated flag is FALSE then the CALL instruction is ignored. Program execution then continues with the next sequential instruction.

RETURN FROM A SUBROUTINE IF THE DESIGNATED FLAG IS TRUE

RTC	043
RTZ	053
RTS	063
RTP	073

These one word instructions will cause a SUBROUTINE to be TERMINATED only if the designated flag is in the logical one (TRUE) state.

CALL A SUBROUTINE IF THE DESIGNATED FLAG IS FALSE

CFC	102
CFZ	112
CFS	122
CFP	132

These instructions are the opposit of the previous group of calling commands. The subroutine is called only if the designated flag is in the FALSE (logical zero) condition. Remember, these instructions must be followed by two words which contain the LOW and HIGH part of the starting address of the SUBROUTINE that is to be executed if the designated flag is FALSE. If the flag is TRUE, the subroutine will not be called and program operation will continue with the next instruction in the current sequence.

RETURN FROM A SUBROUTINE IF THE DESIGNATED FLAG IS FALSE

RFC	003
RFZ	013
RFS	023
RFP	033

These one word instructions will terminate a subroutine ("pop" the program counter stack UP one level) if the designated flag is FALSE. Otherwise, the instruction is ignored and program operation is continued with the next instruction in the subroutine.

THE SPECIAL RESTART SUBROUTINE CALL INSTRUCTIONS

There is a special purpose instruction available that effectively serves as a one word SUBROUTINE CALL. (Remember that it normally requires three words to specify a subroutine call.) This special instruction allows the programmer to call a subroutine that starts at any one of eight specially designated memory locations. The eight special memory locations are at locations: 000, 010, 020, 030, 040, 050, 060 and 070 on page zero. There are eight variations of the machine code for the RESTART instruction. One for each of the above addresses. Thus, the one word instruction can serve to CALL a SUBROUTINE at the specified starting location (instead of having two additional words to specify the starting address of a subroutine). It is often convenient to utilize a RESTART command as a quick CALL to an often used subroutine. Or, as an easy way to call short "starting" subroutines for large programs. Hence, the name for the type of instruction. The eight RESTART instructions, in their mnemonic and machine code forms, along with the starting address associated with each one is listed below.

RST 0	005	00 000
RST 1	015	00 010
RST 2	025	00 020
RST 3	035	00 030
RST 4	045	00 040
RST 5	055	00 050
RST 6	065	00 060
RST 7	075	00 070

INPUT INSTRUCTIONS

In order to receive information from an external device the computer must utilize a group of special signal lines. The typical '8008' computer is designed to handle up to eight groups (each group having eight signal lines) of INPUT signals. A group of signals is accepted at the computer by what is referred to as an INPUT PORT. The computer controls the operation of the INPUT PORTS. Under program control, the computer can be directed to obtain the information that is on a group of lines coming in to any INPUT PORT. When this is done the information will be transferred to the accumulator. Various types of external equipment, such as an electronic keyboard or measuring instruments, can be connected to the INPUT PORTS. The INPUT PORTS are typically referred to as having numbers from '0' to '7.' The typical mnemonics and machine codes for INPUT instructions are shown next.

INP 0	101
INP 1	103
.	.
INP 6	115
INP 7	117

It may be interesting to note that the machine codes for input ports increase by a factor of two for each port. Note too, that while the mnemonic for an input instruction has two parts, the machine code only requires one word in memory. It is also important to realize that while an input instruction brings data into the accumulator it does not affect the status of any of the CPU flags!

OUTPUT INSTRUCTIONS

In order to output information to an external device the computer utilizes another group of signal lines which are referred to as OUTPUT PORTS. A Typical '8008' system may be equipped to service up to twenty-four OUTPUT PORTS. (Each OUTPUT PORT actually consists of eight signal lines.) An OUTPUT instruction causes the contents of the accumulator to be transferred to the signal lines of the designated OUTPUT PORT. The output ports are normally designated by octal numbers in the range 10 to 37. The list below shows the typical mnemonics used to specify an OUTPUT PORT along with the associated machine code. (It may be interesting to note again that the machine code increases by a factor of two for each port.)

OUT 10	121
OUT 11	123
.	.
OUT 21	141
.	.
OUT 36	175
OUT 37	177

An OUTPUT instruction only requires one machine code word (even though the mnemonic is typically specified in two parts). OUTPUT PORTS are connected to external devices that one desires to have the computer transmit information to, such as a CRT display, or machinery that is to be placed under computer control.

THE HALT INSTRUCTION

There is one more instruction in the '8008' instruction set. This instruction directs the CPU to stop all operations and to remain in that state until an INTERRUPT signal is received. In a typical '8008' system an INTERRUPT signal may be generated by an operator pressing a switch or by an external piece of equipment sending an electronic signal to the CPU. This instruction is normally used when the programmer desires to terminate a program or when it is desired to have the computer wait for an operator or external device to perform some action. There are three machine codes that may be used for the HALT command.

HLT	000
HLT	001
HLT	377

The HALT instruction does not affect the status of the CPU flags.

INFORMATION ON INSTRUCTION EXECUTION TIMES

When programming for "real-time" applications it is important to know how much time each type of instruction requires to be executed. With this information the programmer can develop "timing loops" or determine with substantual accuracy how much time it will take to perform a particular series of instructions. This information is especially valuable when dealing with programs that control the operations of external devices which might require events to occur at specific times.

The following table provides the nominal instruction execution time for each category of instruction used in an '8008' system. The precise time needed for each instruction depends on how close the master clock has been set to a nominal value of 500 kilohertz. The table shows the number of cycle states required by the type of instruction followed by the nominal time required to perform the entire instruction. Since each state executes in four microseconds, the total time required to perform the instruction as shown in the table was obtained by multiplying the number of states by four microseconds. By knowing the number of states required for each instruction the programmer can often rearrange an algorithm or substitute different types of instructions to provide programs that have events occuring at precisely timed intervals.

INSTRUCTION EXECUTION TIME TABLE

LOAD DATA FROM A CPU REGISTER TO ANOTHER CPU REGISTER	5	20 Us
LOAD DATA FROM A CPU REGISTER TO A LOCATION IN MEMORY	7	28
LOAD DATA FROM MEMORY TO A CPU REGISTER	8	32
LOAD IMMEDIATE DATA INTO A CPU REGISTER	8	32
LOAD IMMEDIATE DATA INTO A LOCATION IN MEMORY	9	36
INCREMENT OR DECREMENT A CPU REGISTER	5	20
ARITHMETIC/COMPARE BETWEEN ACCUMULATOR & A CPU REGISTER	5	20
ARITH/COMPARE BETWEEN ACCUMULATOR & A WORD IN MEMORY	8	32
IMMEDIATE.ARITHMETIC AND COMPARE	8	32
BOOLEAN OPS BETWEEN ACCUMULATOR AND CPU REGISTERS	5	20
BOOLEAN OPS WITH ACCUMULATOR & A WORD IN MEMORY	8	32
IMMEDIATE BOOLEAN OPERATIONS	8	20
ROTATE THE ACCUMULATOR	5	20
JUMP AND CALL COMMANDS (UNCONDITIONAL)	11	44
JUMP/CALLS WHEN CONDITION NOT SATISFIED (CONDITIONAL)	9	36
JUMP/CALLS WHEN CONDITION SATISFIED (CONDITIONAL)	11	44
RETURN (UNCONDITIONAL)	5	20
RETURN WHEN CONDITION NOT SATISFIED (CONDITIONAL)	3	12
RETURN WHEN CONDITION SATISFIED (CONDITIONAL)	5	20
RESTART COMMAND	5	20
OUTPUT COMMAND	6	24
INPUT COMMAND	8	32
HALT COMMAND	4	16

Chapters 2 and 3 of **MACHINE LANGUAGE PROGRAMMING FOR THE "8008" (and similar microcomputers)** will appear in BYTE's August and September issues, respectively.■

The "Ignorance Is Bliss" Television Drive Circuit

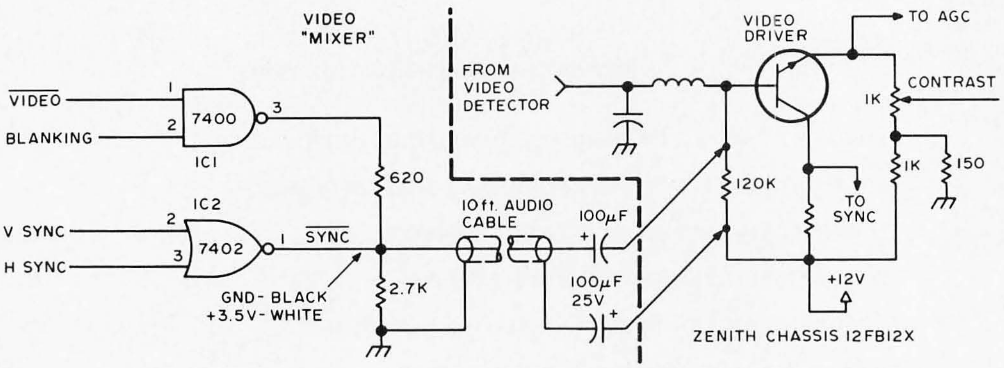


Figure 1: The "Ignorance Is Bliss" Television Drive Circuit. The components to the left of the dashed line were added as part of the interface. The components to the right of the dashed line are part of the Zenith 12FB12X chassis which was used for the television display.

Ken Barbier
PO Box 1042
Socorro NM 87801

I had not yet heard of BYTE magazine, or hams building such hardware, when I built my CRT terminal (a computerhead's term for "TV typewriter") in the fall of 1975. I didn't even own a TV set! Two situations resulted: I had to buy a new all solid state TV, and I didn't have any idea how to interface with it. I knew approximately what it took to create horizontal and vertical sync, but had no idea whether levels, pulse widths, and frequencies would be noncritical. I was delighted, therefore, when my sync generator worked just fine the first time I patched its output across the video driver base resistor using the circuit as shown in figure 1. My big fat TTL level pulses swamp the AGC circuity so effectively that normal signals and noise from the TV IF just disappeared and I had nothing to switch off!

Not having any idea how to mix in my video (character generator output) with the sync, I just hooked up a 2 K pot where the 620 ohm is shown, and started reducing video until it stopped interfering with the sync, and there I was at 620 ohms. All was fine, until I erased my character memory and started typing in one character at a time. Contrast went all to pot! I had provided no DC restoration. And I never did. At least not in the TV set circuitry.

My terminal design produces 24 lines of 64 characters each, with a total of 270 scan lines per frame. Vertical sync is the 10 scan lines that would have been character line 26. To eliminate the need for the type of DC restoration as detailed in "Television Interface" (page 20, BYTE, October 1975), I generated a black level blanking signal covering what would have been character lines 25 thru 27. This signal enters the blanking gate, IC1, at pin 2 in figure 1. Now, when I turn on my system and erase the memory, my TV field shows a nice white area with a black border top and bottom. My character generator output produces black on white characters which I find preferable to the usual white on black.

Simply by turning off the logic power I can be instantly flooded with the inanities emanating from the vast TV wasteland. With this design, I have no need to pull plugs or throw switches. Sometimes ignorance can be bliss.■

Tool Box

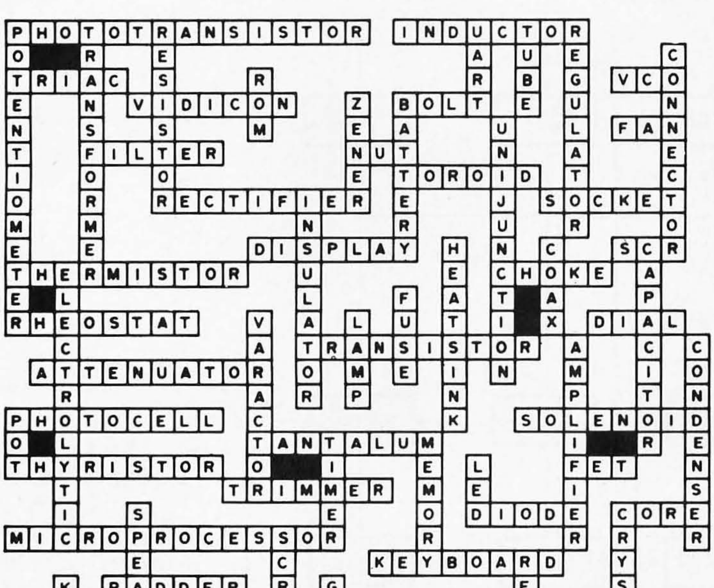
Robert Baker
34 White Pine Dr
Littleton MA 01460

How many different tools and fasteners can you find hidden in the matrix? The letters of each tool name are in a straight line, going horizontally, vertically or diagonally, either backwards or forwards. See how many words you can find "open loop." The official list will appear in next month's BYTE.

H	S	U	R	B	W	M	W	R	A	E	H	S	O	R
A	J	P	E	A	I	I	I	R	E	M	M	A	H	U
L	D	L	A	W	R	C	R	V	I	T	A	P	I	L
I	R	I	M	L	E	R	E	H	I	V	T	A	Y	E
G	I	E	E	T	C	O	S	C	A	S	E	U	H	R
N	L	R	R	W	R	M	T	N	S	R	E	T	C	A
M	L	F	S	E	I	E	R	U	R	E	D	N	E	U
E	O	O	A	E	M	T	I	P	O	V	P	I	R	Q
N	O	R	W	Z	P	E	P	R	S	I	A	B	E	S
T	T	C	R	E	E	R	P	E	S	R	R	B	D	C
S	O	E	E	R	R	U	E	T	I	D	W	L	L	R
E	T	P	N	S	N	A	R	N	C	T	E	E	O	I
K	O	S	C	C	F	I	L	E	S	U	R	R	S	B
I	M	E	H	T	A	L	X	C	K	N	I	F	E	E
D	N	E	E	D	L	E	N	O	S	E	W	O	T	R

Components and Parts Answer

Here is the solution to the Components and Parts puzzle which appeared on page 64 of the June 1976 BYTE.



Put the "Do Everything" Chip in Your Next Design

Robert Baker
34 White Pine Dr
Littleton MA 01460

The TMS-5501 is a multifunction, IO controller available from Texas Instruments in a 40 pin dual in line package. It provides an asynchronous communications interface with programmable data rate, a parallel data IO buffer, interrupt control logic, and five programmable interval timers in a single chip. Although designed specifically for use with a TMS-8080 central processor, it may be used with almost any other microprocessor as described in this article. Figure 1 shows a functional block diagram of the TMS-5501 and table 1 lists the pin assignments along with a brief description of each pin.

The system data bus is used for data transfers between the TMS-5501 and the microprocessor as controlled by the interrupt, chip enable, sync, and address lines. A convenient method of addressing the TMS-5501 is to connect the chip enable to the high order address line and the four TMS-5501 address inputs to the four lower order address bits of the address bus, limiting the system to 32 K bytes of memory space (on an 8080 based system). Alternately, a full address decode could be done with additional hardware logic. The four memory address inputs of the TMS-5501 are then used to select one of the 14 possible commands shown in table 2. The individual commands are actually generated by executing memory reference instructions, with the low order hexadecimal memory address being the TMS-5501 command. This provides great flexibility by allowing the use of any memory reference instruction for IO operations.

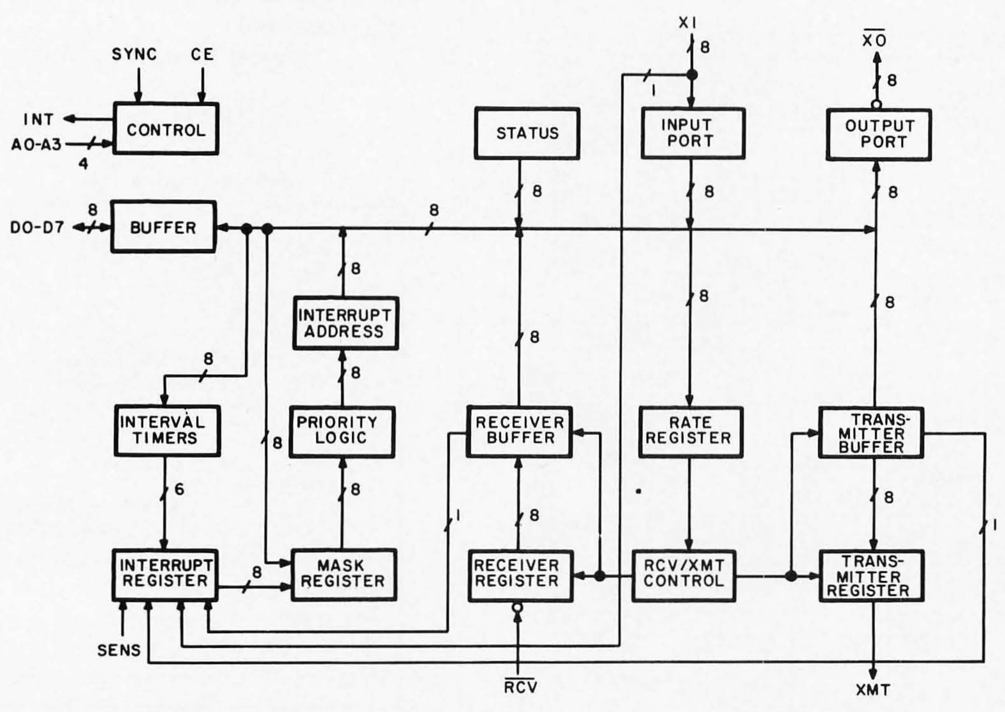


Figure 1: Block Diagram of the Texas Instruments TMS-5501 Design. This diagram shows the elements which are contained within this "do everything" IO chip. Notations next to the interconnection lines specify the number of bits involved in the data path.

Table 1: TMS-5501 Pin Assignments and Functions. This information is taken directly from the Texas Instruments Incorporated TMS-5501 Multifunction Input/Output Controller Manual.

SIGNATURE	PIN	DESCRIPTION
		INPUTS
CE	18	Chip enable – When CE is low, the TMS 5501 address decoding is inhibited, which prevents execution of any of the TMS 5501 commands.
A3	17	Address bus – A3 through A0 are the lines that are addressed by the TMS 8080 to select a particular TMS 5501 function.
A2	16	
A1	15	
A0	14	
SYNC	19	Synchronizing signal – The SYNC signal is issued by the TMS 8080 and indicates the beginning of a machine cycle and availability of machine status. When the SYNC signal is active (high), the TMS 5501 will monitor the data bus bits DO (interrupt acknowledge) and D1 ($\overline{\text{WO}}$, data output function).
$\overline{\text{RCV}}$	5	Receiver serial data inputline – $\overline{\text{RCV}}$ must be held in the inactive (high) state when not receiving data. A transition from high to low will activate the receive circuitry.
X10	39	External inputs – These eight external inputs are gated to the data bus when the read-external-inputs function is addressed. External input n is gated bus bit n without conversion.
X11	38	
X12	37	
X13	36	
X14	35	
X15	34	
X16	33	
X17	32	
SENS	22	External interrupt sensing – A transition from low to high at SENS sets a bit in the interrupt register, which, if enabled, generates an interrupt to the TMS 8080.
		OUTPUTS
$\overline{\text{XO}}$ 0	24	External outputs – These eight external outputs are driven by the complement of the output register; i.e.: if output register bit n is loaded with a high (low) from data bus bit n by a load-output register command, the external output n will be a low (high). The external outputs change only when a load-output-register function is addressed.
$\overline{\text{XO}}$ 1	25	
$\overline{\text{XO}}$ 2	26	
$\overline{\text{XO}}$ 3	27	
$\overline{\text{XO}}$ 4	28	
$\overline{\text{XO}}$ 5	29	
$\overline{\text{XO}}$ 6	30	
$\overline{\text{XO}}$ 7	31	
XMT	40	Transmitter serial data output line – This line remains high when the TMS 5501 is not transmitting.
		DATA BUS INPUT/OUTPUT
D0	13	Data bus – Data transfers between the TMS 5501 and the TMS 8080 are made via the 8 bit bidirectional data bus. D0 is the LSB. D7 is the MSB.
D1	12	
D2	11	
D3	10	
D4	9	
D5	8	
D6	7	
D7	6	
INT	23	Interrupt – When active (high), the INT output indicates that at least one of the interrupt conditions has occurred and that its corresponding mask-register bit is set.
		POWER AND CLOCKS
V_{SS}	4	Ground reference
V_{BB}	1	Supply voltage (–5 V nominal)
V_{CC}	2	Supply voltage (5 V nominal)
V_{DD}	3	Supply voltage (12 V nominal)
$\phi 1$	20	Phase 1 clock
$\phi 2$	21	Phase 2 clock

Command Functions

Addressing the "Issue Discrete Command" function (storing data in memory location XXX4) will cause the TMS-5501 to decode the data bus information as shown in figure 2. Bits 1, 2, and 3 are latched until a new discrete command is received. Setting the RESET bit low has no effect, while setting it high will:

1. Clear the receiver flags in the TMS-5501 but not the receiver buffer.
2. Set the transmitter data output high (marking) and set the XMIT BUFFER EMPTY bit high to indicate the buffer is ready to accept a character from the processor.
3. Clear the interrupt register except for the transmitter buffer interrupt.
4. Inhibit interval timers.

Setting the Interrupt Acknowledge Enable bit high will allow the TMS-5501 to decode the processor status (bit 0 of the data bus at SYNC of each 8080 machine cycle) to determine if an interrupt acknowledge is being issued by the processor and initiate appropriate interrupt operation. Otherwise, with the bit set low, the TMS-5501 will ignore the interrupt acknowledge. Bits 4 and 5 are only used for testing purposes and should always be kept low.

Reading the TMS-5501 status (memory reference to address XXX3) will return a data byte indicating the status of the controller as shown in figure 3. A brief description of each bit follows:

Bit 0 – A high indicates a framing error detected on the last received character.

Bit 1 – A high indicates a new character was loaded into the receiver buffer before the previous character was read.

Bit 2 – Normally high when no data is being received.

Bit 3 – A high indicates a new character is in the receiver buffer.

Bit 4 – A high indicates an empty transmitter buffer ready to accept a character.

Bit 5 – A high indicates one or more

Low Order Address (Hexadecimal)	A3	A2	A1	A0	Command	Function
	Binary Address					
0	L	L	L	L	Read receiver buffer	RBn → Dn
1	L	L	L	H	Read external inputs	XIn → Dn
2	L	L	H	L	Read interrupt address	RST → Dn
3	L	L	H	H	Read TMS 5501 status	(Status) → Dn See figure 3
4	L	H	L	L	Issue discrete commands	See figure 2
5	L	H	L	H	Load rate register	See figure 4
6	L	H	H	L	Load transmitter buffer	Dn → TBn
7	L	H	H	H	Load output port	Dn → $\overline{\text{XO}}$n
8	H	L	L	L	Load mask register	Dn → MRn
9	H	L	L	H	Load interval timer 1	Dn → Timer 1
A	H	L	H	L	Load interval timer 2	Dn → Timer 2
B	H	L	H	H	Load interval timer 3	Dn → Timer 3
C	H	H	L	L	Load interval timer 4	Dn → Timer 4
D	H	H	L	H	Load interval timer 5	Dn → Timer 5
E	H	H	H	L	No function	
F	H	H	H	H	No function	

Table 2: TMS-5501 Commands. The commands presented to this device are best implemented by memory references to addresses XXXO through XXXF (hexadecimal) where XXX is an arbitrary location in memory address space. The low order hexadecimal address for each command, along with the binary code presented to the address lines. The notation Dn refers to data bus line "n" where n is 0 to 7.

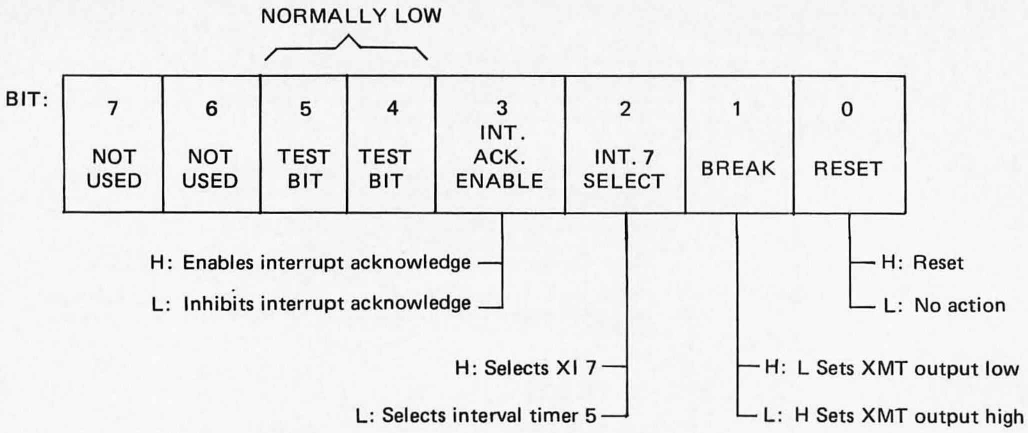


Figure 2: Discrete Command Format. The data bus contents stored in the TMS-5501 when address location XXX4 receives processor data is interpreted using this format.

interrupt conditions have occurred and the corresponding interrupt is enabled.

Bit 6 – A high indicates detection of the first data bit of a receive-data character.

Bit 7 – A high indicates detection of the start bit of an incoming data character.

Addressing the load rate register function (storing data in memory location XXX5) will allow the TMS-5501 to load the internal data rate register from the data bus. The bit assignments of the rate register are illustrated in figure 4. Bit 7 selects the desired number of stop bits while setting a bit between bits 0 and 6 selects the transmitter/receiver baud rate between 110 and 9600 baud. If more than one rate bit is high, the highest rate will be selected. If bits 0 to 6 are all low, the receiver and transmitter circuitry will be disabled.

Reading the receiver buffer (address XXX0) or external inputs (address XXX1) will return the data received from the external devices while loading the transmitter buffer (address XXX6) or output port (address XXX7) will output data to the external devices. The load mask register (address XXX8) function loads the data byte into an interrupt mask register where a high in bit "n" enables interrupt "n". See table 3 for a list of the possible interrupts, their priorities, and information on the corresponding RST instruction for an 8080 processor. Addressing the read-interrupt address function will return the highest priority interrupt address (the RST instruction) and clear the corresponding bit in the interrupt register. The TMS-5501 status should be checked for an interrupt pending before trying to read an interrupt address to avoid false data.

Loading an interval timer loads the contents of the data bus into the corresponding interval timer and activates that particular timer. The timer then counts down in increments of 64 microseconds providing a programmable interval of 64 to 16,320 microseconds with longer intervals generated by cascading timers through software. When a timer reaches zero, it generates an interrupt which in turn can control any desired function such as scanning a keyboard or switch matrix, time slicing in a multiprogramming environment, etc. Loading a new value while a timer is counting overrides the previous value and the timer starts counting down the new value. If a zero value is loaded to a timer, an interrupt is generated immediately.

Interrupt System

The TMS-5501 provides several interrupt control functions by receiving external interrupt signals, generating interrupt signals to the processor, masking out undesired interrupts, establishing priority of interrupts, and generating RST instructions for an 8080 processor. External interrupts are normally received on pin 22, the SENS input, but an additional external interrupt can be received at the X17 input pin if selected by a discrete command. The TMS-5501 will generate an interrupt to the processor whenever any of the five internal interval timers counts to

BIT:	7	6	5	4	3	2	1	0
	START BIT DETECT	FULL BIT DETECT	INTRPT PENDING	XMIT BUFFER EMPTY	RCV BUFFER LOADED	SERIAL RCVD	OVERRUN ERROR	FRAME ERROR

Figure 3: Status Word Format. The data read into the processor when TMS-5501 address location XXX3 is referenced is in this format.

BIT:	7 STOP BIT(s)	6 9600 baud	5 4800 baud	4 2400 baud	3 1200 baud	2 300 baud	1 150 baud	0 110 baud

- H: One stop bit
- L: Two stop bits

Figure 4: Data Rate Command Format. The data bus contents stored in the TMS-5501 address location XXX5 has this format, and is used to control the internal data rate generator for serial communications. A high level on the appropriate selection bit sets that data rate unless another bit at a higher rate is also selected.

zero, the receiver buffer is loaded, or the transmitter buffer is empty. When an interrupt signal is generated, it is compared with the mask register. If that particular interrupt is enabled, it is passed on to the priority logic which allows the highest priority interrupt to generate an RST instruction to the 8080 processor only if no higher priority interrupt is still pending.

Two methods of servicing interrupts are provided: an interrupt driven system utilizing the RST instructions or a polled interrupt system utilizing bit 5 of the TMS-5501 status byte (the interrupt pending bit). In an interrupt driven system, the INT output signal of the TMS-5501 is connected to the INT input of an 8080 processor. The normal interrupt sequence would be as follows:

1. TMS-5501 receives an external interrupt or generates an internal interrupt signal and sets the appropriate RST instruction.
2. The INT output goes high signaling the processor that an interrupt has occurred.
3. If the 8080 is enabled to accept interrupts, it sets the INTA (interrupt acknowledge) status bit high at SYNC time of the next machine cycle.
4. If the TMS-5501 has previously received an interrupt acknowledge enable command from the processor (see bit 3 of discrete command), the RST instruction will be transferred on the data bus to the CPU.

In a polled interrupt system, the INT output is not used and the interrupt sequence would be as follows:

	Data Bus Bit								
Priority	0	1	2	3	4	5	6	7	Interrupt Caused By
1	H	H	H	L	L	L	H	H	Interval Timer 1
2	H	H	H	H	L	L	H	H	Interval Timer 2
3	H	H	H	L	H	L	H	H	External Sensor
4	H	H	H	H	H	L	H	H	Interval Timer 3
5	H	H	H	L	L	H	H	H	Receiver Buffer
6	H	H	H	H	L	H	H	H	Transmitter Buffer
7	H	H	H	L	H	H	H	H	Interval Timer 4
8	H	H	H	H	H	H	H	H	Interval Timer 5 or X17

Table 3: Interrupt Assignments. This table lists the priority level and source of each interrupt. For 8080 systems without previously dedicated restart instructions, the TMS-5501 can be used to automatically generate RST n where n is the priority level. Otherwise, some form of polling is required.

1. The TMS-5501 receives or generates an interrupt and sets the corresponding RST instruction.
2. The TMS-5501 interrupt pending status bit (see bit 5 of status byte) is set high.
3. At some predetermined time, the processor polls the TMS-5501 by reading the status byte and checking the interrupt pending bit.
4. If the bit is high, the processor should then read the interrupt address which will cause the TMS-5501 to transfer the RST instruction to the processor as data.
5. The software can then decode the RST instruction to determine the interrupting device.

With an 8080 processor, the TMS-5501 controller may be used in either interrupt mode. For any other microprocessor based system, the TMS-5501 would be easiest to implement using a polled interrupt system that would not require any extra hardware to interface with the CPU. An alternate method would use part of both interrupt schemes by connecting the INT output to the CPU interrupt input but disabling the TMS-5501 interrupt acknowledge enable bit (bit 3 of discrete commands = low). This would eliminate the need to poll the TMS-5501 status byte but the interrupt address (the RST instruction) would still have to be read and decoded by the CPU.

Applications

The TMS-5501 appears to be an ideal interface for the typical personal computer system with many different possible applications. For example, suppose we connected a CRT to the external output port and an ASCII keyboard to the external input port. The serial IO port could then be used with a Teletype for hardcopy, an accoustic coupler for loading from another system (or intersystem communications), or for a cassette drive. You may want to connect a switch to the SENS input to cause an interrupt that would return to a system control or debug routine.

For an ideal cassette interface, the serial IO could transfer data to and from the cassette deck while the external outputs could be used to control drive functions such as forward, reverse, fast forward, etc. The external inputs might be used for inputting status information from the drive such as end of tape, start of tape, tape jam, etc. With this scheme, true searches and tape scans could be accomplished under full software control with no manual actions required once a tape is inserted in the drive.

As you can clearly see, there's an endless number of possibilities for the TMS-5501 limited only by your imagination. Even if your system is not based on an 8080 processor, it is well worth looking at the TMS-5501 for your IO needs.■

BYTE'S BITS

Election Program

The following is paraphrased from a version submitted by Mark T O'Bryan of Portage MI, who credits it to a friend at MIT.

Question: "What's the biggest problem in writing

a { Jerry Ford / Scoop Jackson / Jimmy Carter / Ronald Reagan } simulator for an Altair with only 4096 bytes of memory?

.
.
.

Answer: Trying to figure out what to do with the other 4095 bytes of memory.■

New Hope For Computer Phreaques

According to a report published in *Electronic Products* magazine, April 1976, page 16, Commander Thomas Orr of Warsash, England, has created a digital electronic watch which reads the pulse rate of its wearer. Just think, now you can check to see if you're dead or not. The watch is marketed in the United States by Pulse Watch of Tiburon CA and the transducer technology on which it is based has much wider applications in medical electronics.■



Now, you can buy an Altair 8800 or Altair 680 computer kit right off the shelf. Most all Altair options, software and manuals are also available. The MITS Dealer List below is just the beginning:



off the shelf

RETAIL COMPUTER STORE, INC.
410 N. E. 72nd
Seattle, WA 98115
(206) 524-4101

CTI DATA SYSTEMS
3450 East Spring St.
Long Beach, CA 90806
(213) 426-7375

COMPUTER KITS
1044 University Ave.
Berkeley, CA 94710
(415) 845-5300

GATEWAY ELECTRONICS
2839 W. 44th Ave.
Denver, CO 80211
(303) 458-5444

ARROWHEAD COMPUTER CO.
(THE COMPUTER STORE)
820 Broadway
Santa Monica, CA. 90401

GATEWAY ELECTRONICS
8123-25 Page Blvd.
St. Louis, MO 63130
(314) 427-6116

BYTE'TRONICS
5604 Kingston Pike
Knoxville, TN 37919
(615) 588-8971

THE COMPUTER SYSTEMCENTER
3330 Piedmont Road
Atlanta, GA 30305
(404) 231-1691

MARSH DATA SYSTEMS
5405-B Southern Comfort Blvd.
Tampa, FL 33614
(813) 886-9890

the COMPUTER STORE, INC.
120 Cambridge
Burlington, Mass. 01803
(617) 272-8770

MICROSYSTEMS
6605A Backlick Rd.
Springfield, VA 22150
(Washington DC area)
(703) 569-1110

THE COMPUTER ROOM
3938 Beau D'Rue Drive
Eagan, Minn. 55122
(612) 452-2567

COMPUTER PRODUCTS UNLIMITED
4216 West 12th St.
Little Rock, Arkansas 72204
(501) 666-2839

NOTE: Altair is a trademark of MITS, Inc.

Why Wait?

Build a *FAST* Cassette Interface

Dr Robert Suding
Research Director for Digital Group Inc
PO Box 6528
Denver CO 80206

This cassette interface does not have a ±30% speed tolerance. The design requires ±12 V and +5 V to run. A good quality recorder must be used, along with excellent quality tapes. Careful adjustments are required.

So why use it? Well, it works! It's dependable. And it's fast. In contrast, the proposed BYTE standard cassette interface runs at 300 Baud. A Teletype paper tape reads @ 110 Baud. I have 24 K on my system. How long would it take me to completely load my system (not including any Bootstrap Loader operations)?

Teletype @ 110 Baud – 40 minutes 58 seconds

Proposed BYTE standard @ 300 Baud – 15 minutes 1 second

The system to be shown in this article has been running for almost a year at 1100 Baud (with an upper limit of 1750 Baud with critical tuning).

Suding system @ 1100 Baud – 4 minutes 6 seconds

Past issues of BYTE have included several articles on cassette interface proposals and circuits. I would suggest re-reading these articles. You will find one common element. Slow. If you get the impression that I'm impatient, you're right. I'll bet you are too. Imagine reading 300 Baud for 15 minutes to discover a noise pulse had destroyed data, requiring re-reading. Ugh!

Thus the proposed standard of the BYTE Kansas City conference in 1975 has a major disadvantage: The use of a redundant Manchester format with a 1200 Hz low frequency critically restricts the user to slower data rates. A related disadvantage for those who use filters or phase lock loops as an input detection method is the fact that the Manchester code employs harmonically related frequencies; this leads to design problems in detectors based upon frequency discrimination techniques.

The system shown in this article avoids the above pitfalls. It uses the non-harmonically related tones of 2125 Hz – Mark and 2975 Hz – Space. The second harmonic of 2125 Hz occurs at 4250 Hz, well down on the passband of a 2975 Hz detector. Sufficient space exists between the two frequencies to allow for reasonable recorder speed discrepancies. The higher frequencies involved permit increasing the data rate.

Several approaches are possible in cassette interfacing, as seen in past BYTE articles. However, their emphasis on wide cassette speed tolerance made them slower. My

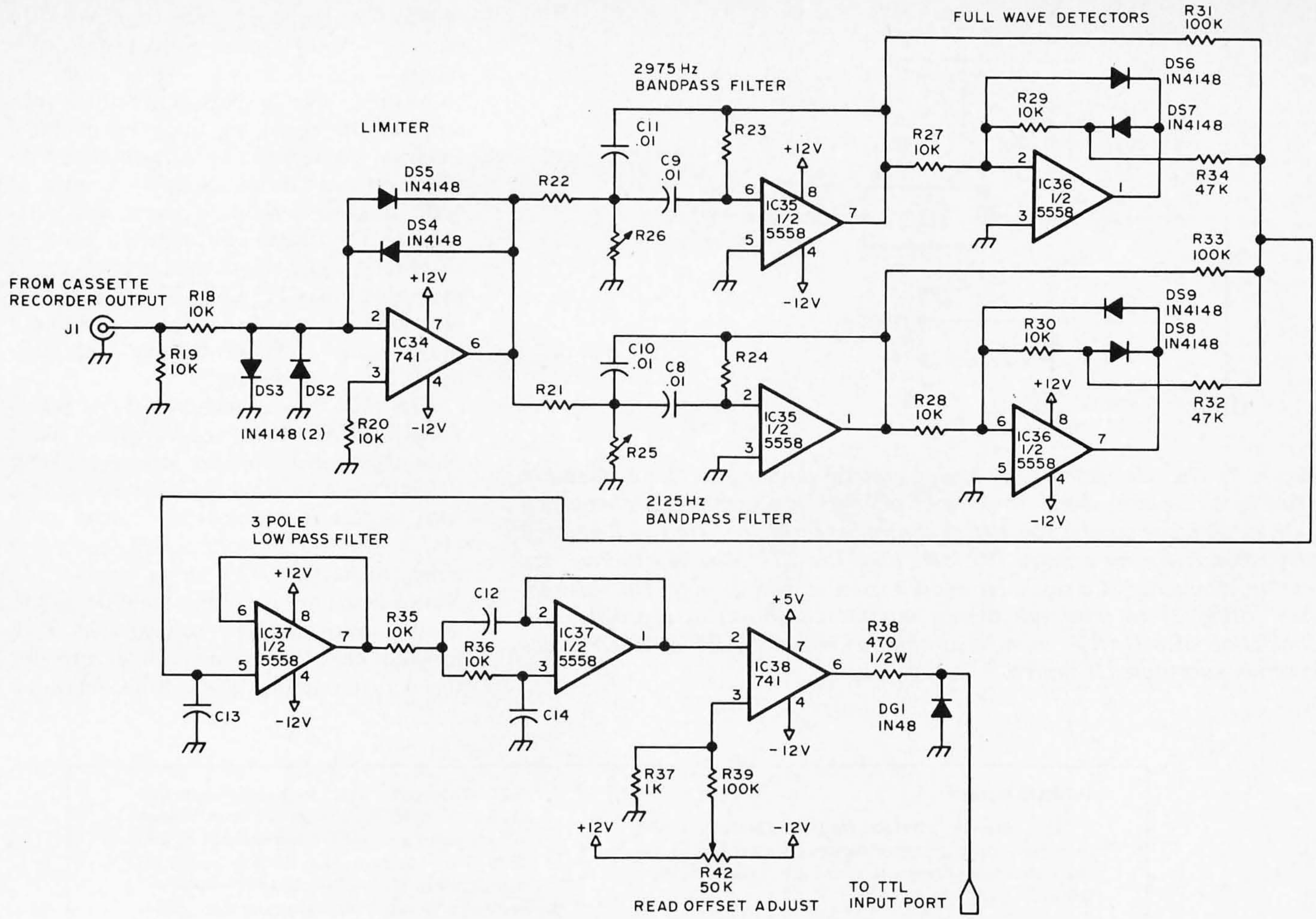


Figure 1: The schematic of the Suding cassette input interface as found in the Digital Group systems. This interface amplifies and clips the cassette output with limiting amplifier IC34, discriminates the two data frequencies (see table 1) with bandpass filters followed by full wave detectors, passes the detected signal through a 3 pole active low pass filter, then converts the result to a TTL level which is read by a single bit input port. One example of software (see listing 1) to drive this input interface uses a programmed simulation of UART input algorithm; an actual UART or ACIA device could be substituted if desired.

approach to "out of specification cassette speed" is – "put it in the specification, or get a good recorder." More of that later.

Theory of Operation

The 1100 Baud Digital Group system uses the circuits of figures 1 and 2. The cassette receive circuitry detects the prerecorded frequency shift keying and produces a "1" or a "0" output as a result of a detected 2125 Hz or 2975 Hz tone at the input. A 741 operational amplifier, IC34, is used as a clamped limiter which prevents variations in cassette amplitude from affecting the detection process. The output of the limiter should be about .6 V peak to peak, roughly a square wave with rounded edges of the incoming frequency, constant in amplitude regardless of tape volume setting or minor tape "dropout" problems.

Two bandpass active filters (IC35) then amplify a tone five times when actually tuned to their respective frequencies of 2975 Hz for the top filter, and 2125 Hz for the lower filter. The further off the tuned frequency the tone is, the less amplification the filter will produce. The gain, bandwidth, and tuned frequency are set by the three resistors and two condensers in each filter. Each filter may be exactly tuned to frequency by carefully setting the variable resistance value (which may be either a potentiometer or selected fixed values).

Full wave active detectors produce rectified full wave pulses at the summing junction, pin 5 of IC37. The 2975 Hz tones are rectified to a positive voltage, and the 2125 Hz tones are rectified to a negative voltage. As received tones depart from either exact frequency, a value less positive or

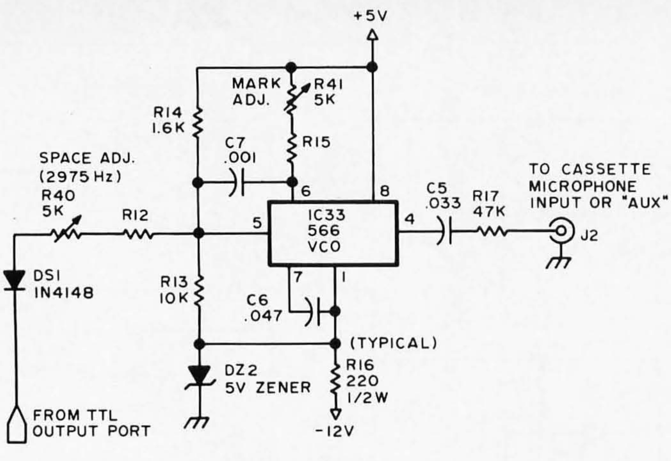


Figure 2: The schematic of the Suding cassette output interface as found in the Digital Group systems. The output interface is a simple audio frequency shift keyer made up of a 566 voltage controlled oscillator with two frequency states controlled by a single TTL data line. The TTL level which drives the output modulator is a single bit derived from an output port. The software (see listing 2) to drive this output interface is shown as a programmed simulation of a UART output algorithm; an actual UART or ACIA device could be substituted if desired.

negative is produced until approximately midway (2550 Hz) a summed voltage of 0 results.

A three pole lowpass active filter then removes the remaining traces of pulsating DC from the summed signal with almost no effect on the data pulses up to a speed of 1000 bits per second. If lower data rates were to be utilized, an improved signal to noise ratio could be obtained by multiplying the values of C12, C13, and C11 by the reciprocal of the data rate ratio. Table 1 shows some component values for alternative frequency designs.

The final receiver section is a 741 operational amplifier, IC38, connected as a slicer. This operational amplifier detects whether the voltage at its pin 2 is positive or negative with respect to the constant voltage at its pin 3. The output voltage will then swing either to nearly −12 V or to nearly +5 V. Notice that this operational amplifier has +5 as its positive supply voltage, pin 7. A forward biased germanium diode prevents the actual output voltage from going less

Tune Up Notes

The cassette interface must be carefully tuned to achieve proper performance. Careless tuning has been the most frequent cause of cassette system failure.

1. Plug in the six integrated circuits of the cassette interface.
2. Connect a calibrated audio oscillator between the limiter input and ground. A digital frequency counter driven by the audio oscillator is highly recommended. The oscillator should cover the desired range of 2 – 3 kHz, with a sine wave output of .5 or so, although the precise level is not at all critical.
3. Apply +5 and ±12 voltages to the circuit. Measure the output at pin 6 of the 741 limiter (IC34) with an oscilloscope. The wave shape should be a rounded square wave of about .6 V peak to peak.
4. Set the audio oscillator to 2125 Hz. Measure the output at pin 1 of the 5558 active bandpass filter. Slowly turn R25 until the signal peaks. Be sure that you are peaking at 2125 Hz, not a harmonic. Vary the oscillator frequency a few decades to insure 2125 Hz is the tuned frequency.
5. Similarly, set the oscillator to 2975 Hz and measure the output at pin 7 of the 5558 (IC35). Slowly turn R26 until the signal peaks. Vary the oscillator to insure a 2975 Hz peak.
6. Measure the detected voltages at pin 5 of IC37. When the oscillator approaches 2125, the voltage should go negative. When approaching 2975, the voltage should go positive. Trouble in this area would most likely be caused by reversed or defective diodes, or shorts between adjacent lines.
7. Measure the voltage at the cathode (bar) end of the output clamping germanium diode (G1). Sweeping the frequency between 2125 and 2975 Hz should result in a clean voltage jump somewhere between 2125 and 2975 Hz. Measure the output swing to insure that it does not exceed +5, –.3 V.
8. Remove the audio oscillator and short input connector J1 temporarily to ground. Measure the output at pin 6 of IC34. A stable condition (no oscillation) should be seen. Connect the oscilloscope to the cathode of G1 again. Adjust the balance potentiometer (R42) so that the output voltage is a negative level. Slowly turn the potentiometer until the output voltage jumps to a positive level and leave the setting at this point.
9. Disconnect the temporary jumper from the input connector and reconnect the audio oscillator. Perform step 7 again. The crossover threshold should be close to 2550 now. If all proceeds well at this point, the cassette interface is ready to receive data.
10. Connect the oscilloscope to pin 4 of the 566 voltage controlled oscillator (IC33). A triangular wave output should be seen.
11. Connect a temporary jumper between the TTL input going to DS1 and +5 V. Connect a frequency counter to pin 3 of the VCO (IC33). Adjust potentiometer R41 for a resultant output frequency of 2125 Hz.
12. Remove the jumper from +5 V and connect the jumper from DS1's input to ground. This time adjust R40 for 2975 Hz output.
13. Remove the jumpers, and you are ready for final tune in the driving circuit. Connect the cassette interface to the driving output port, and program the driving processor to send a TTL high level ("1") output to the cassette interface. Adjust R41 to 2125 Hz. Then have the processor send a "0" level. This time adjust R40 for 2975 Hz output. The cassette interface is now ready for use.

than ≃ −.2 V, so that valid TTL levels are not exceeded. An offset adjusting potentiometer allows the output to be placed in a "Mark Hold" condition when no tone input is being detected.

The cassette recording section (figure 2) uses a single integrated circuit, a 566 voltage controlled oscillator, IC33. A logic level from the computer's output port controls the resultant audio frequency output to the cassette recorder microphone input. A high input ("1") produces a 2125 Hz output, and a low input ("0") results in 2975 Hz. The output wave shape is a symmetrical triangular wave. Should the user object to using a triangular wave, a more nearly sine wave can be obtained by connecting a pair of back to back 1N914 diodes between ground and the output side of the coupling capacitor C5.

Exact values and high quality components will result in a trouble-free voltage controlled oscillator. The 47 K (R17) resistor in series with the output is a typical value to be used when coupling to the low level, low impedence external microphone inputs of most cassette recorders. Using the "AUX" input of your cassette recorder generally gives better results.

Construction

The cassette interface is available as a part of a printed circuit board kit from the Digital Group. The printed circuit board is shared by a television display circuit to be described in the next article in this series. A kit of the cassette interface only is also available from the Digital Group for $30, which includes all parts and the printed circuit board. The experienced builder can build the circuit in an evening or two by hand wiring components on standard .1 inch grid Vectorboard. All the circuitry can be contained in an area of approximately 3 inch by 5 inch (about 8 cm by 13 cm).

Be sure to use only high quality components, particularly in the active bandpass filters and voltage controlled oscillator. Some strange "frequency jump" problems have been traced to surplus 566s which were temperature sensitive. Lay out the receive circuit to avoid feedback paths from output to input, particularly in the limiter, active bandpass filters, and slicer areas. Different op amps could be used, but may result in instability or degradation of final performance due to suboptimization.

Modifying Your Cassette Recorder

It is very helpful to listen to the data from the cassette so that the beginning of the data burst may be detected, as well as

	Low Filter			High Filter			Low Pass Filter			VCO	
	R21	R24	R25 *	R22	R23	R26 *	C13	C12	C14	R12	R15
2125-2975 Hz 1100 Baud	6.8 k	68 k	938	4.7 k	47 k	697	.0056 μF	.01	.015	2.7 k	1.3 k
1200-2400 Hz 300 Baud (Simple)	6.8 k	68 k	4173	4.7 k	47 k	1162	.0056 μF	.01	.015	470 k	2.7 k
1200-2400 Hz 300 Baud (Correct)	12 k	120 k	1668	5.6 k	56 k	906	.015 μF	.033	.047	470 k	2.7 k
2125-2295 100 Baud (Simple)	6.8 k	68 k	938	4.7 k	47 k	1301	.0056 μF	.01	.015	47 k	2.7 k
2125-2295 100 Baud (Correct)	36 k	360 k	156	27 k	270 k	179	.056 μF	.1	.15	47 k	2.7 k

*** means that the value so indicated is the typical calculated value. The precise value is dependent on component tolerance.**

Table 1: Theoretical values of components for alternate frequencies. This table gives values of components to be used with the circuits of figures 1 and 2 in order to make this cassette interface work with several alternate specifications. See the text for a definition of the various comments at the left of the table.

Potential Troubles

Knowing about potential problem areas is a first step to minimization of their effects. Troubles seem to break down into *six* classes.

- Cassette recorders and the cassettes used: A marriage between your $1000 microprocessor and junior's $20 cassette recorder, which has been using 30¢ cassettes for the last five years, will not produce happy offspring! I have been using a Superscope C-104 for the past year, and can report no failures except for defective cassette tapes. The C-104 has several attractive features. Besides the usual conveniences such as index counter, cuing, etc, it has a variable readback speed control, dandy for out of spec cassettes from friends. Inside, another special motor speed control potentiometer is located near the speaker which allows precisely setting the record/write speed. Quality control seems good overall, and the list price of $120 (cheaper at discount stores) is worth the investment. Don't waste your money on cheap cassettes. Sony Low Noise C-45s have been generally good. Some $2 – $4 Data Certified Cassettes are superior, but not needed.
- Microprocessor caused problems: Some microprocessor designs will not work directly with this interface system. This interface was designed to be connected directly to a single bit IO port, with the processor handling all of the bit timings through timing loops. If your processor must periodically catch its breath for such things as dynamic memory refreshing, you may be unable to directly use the "Software UART" system. What a shame! However, a hardware UART will permit using the system even with a system of this nature.
- Cabling problems: It is possible to connect your cassette recorders with the read and write cables reversed. Enough crosstalk from the write line to the read limiter existed to give the appearance of data being read, but so many errors resulted that the programming would not run.
- Tuning problems: Circuit tuning is the most common problem. *Carefully* tune the active filters!
- Cassette Crashes: Cassette damage is frequent on tapes which have always worked before, but now mysteriously fail. The most common cause of this is removing a cassette from the recorder without completely rewinding. The exposed oxide then gets damaged, and is no longer usable.
- Miscellaneous circuit problems:

Defective level output from cassette read limiter.
1. None at all: Check for ±12 V to IC34, and IC34.
2. Too high output level: Diodes (DS4 and DS5) open, or one is reversed.

Bandpass active filters don't filter.
1. Off frequency
2. Bad 5558
3. Check for shorts or out of tolerance condensers C8, C9, C10, or C11. Disk ceramics are a "no-no" in tuned circuits.
4. Resistors improperly wired or inserted.

Full wave detector does not work as described:
1. Diodes open, reversed or shorted.
2. Defective IC36.

Low pass active filter fails to work:
1. Shorted or out of tolerance condensers.
2. Defective IC37.

Output slicer (IC38) fails to produce TTL levels:
1. Reversed, open or not Germanium diode at DG1.
2. Too heavily loaded output. This circuit should drive no more than one TTL load (standard for most IO ports).

VCO won't oscillate.
1. Defective 566 (IC33).
2. Shorted condenser C6.

VCO has parasitic oscillation (high frequency):
1. C7 not connected.
2. Defective 566.
3. C6 is open, producing a very high frequency.

VCO won't tune to frequency or stay there:
1. Out of tolerance or defective C6. You really didn't use a disk ceramic here, did you?
2. Defective 566.
3. Non-TTL levels used to drive VCO.
4. Defective potentiometers R40 or R41.
5. DS1 or DZ2 reversed or defective.

hearing the end of the data. When the cassette read cable is plugged into most cassette recorders' earphone output jack, the speaker output is usually cut off. However, since a closed circuit jack is all that is involved, a quick solution is to connect a jumper on the jack so that the speaker is not disconnected. Even better, use a 100 ohm ¼ watt resistor instead of the jumper, and the data howl won't be so loud. A 10 ohm, ¼ watt resistor from the amplifier lead to jack, to the jack frame will prevent potential damage to the output driving transistor(s).

Alternative Frequencies and Applications

The cassette interface design may be used with the proposed BYTE standard should you so desire. Table 1 has appropriate component values calculated for two alternative possibilities: the simple way (less desirable) and the "right way". The simple way permits using a switch on the bandpass active filters to select the frequency pairs. The right way involves setting the circuit to the optimal values, and using separate interfaces for each frequency pair.

Amateur radio (ham) radioteletype (RTTY) generally uses 2125 – 2295 Hz frequency shift keying for 170 Hz shift. The existing cassette interface can be used by "straddle tuning," but improved performance may be obtained by selecting a second R26 which will tune the high filter to 2295. The cassette read cable may then be attached to the short wave receiver and the microprocessor, programmed as a radioteletype video terminal, which can replace the noisy Teletype machine. Of course, a cassette interface specifically designed for this 170 Hz shift at 100 WPM will give superior performance under marginal conditions.

The cassette interface may be used as a stand alone radioteletype terminal unit and audio frequency shift keying if desired, and works quite nicely in this application.

Software

I would suggest using software for your cassette read and write timings. Sample 8080 software is included as listing 1. Timings at locations <0>/116, <0>/133, <0>/241, and <0>/260 are based on an 8080 system with a 500 ns T time and no wait states. Slower systems will require proportionately decreased loop timings.

A UART could be used instead of the "software UART" system shown. However, several disadvantages arise. First, a slightly greater cost and complexity. More important, however, is a degradation in total

Listing 1: Stand Alone Suding Cassette Input Program. This program is a self contained data transfer routine which will transfer a block of data from cassette to split octal memory locations xxx/xxx through yyy/000. This program assumes that MEMTOCAS (see listing 2) was used to create the tape being read. A more generally useful input facility would be modelled on this program and linked to a system monitor as a subroutine.

Split Octal Address	Octal Code	Label	Op.	Operand	Commentary
<0>/100	041 xxx xxx	CASTOMEM	LXI	H,xxx/xxx	Load starting address in HL pair;
<0>/103	021 010 000	STARTBYT	LXI	D,000/000	Load E, clear D;
<0>/106	333 001	SYNCHLOO	IN	1	Port 1 bit 0 read for input;
<0>/110	346 001		ANI	1	Mask all but bit 0;
<0>/112	302 106 <0>		JNZ	SYNCHLOO	If not start bit then reiterate loop;
*<0>/115	006 300		MVI	B,300	Time delay to middle of first data bit*;
<0>/117	005	WSYNCH	DCR	B	Decrement synch wait count;
<0>/120	302 117 <0>		JNZ	WSYNCH	If not done then keep waiting;
<0>/123	333 001	GETDATA	IN	1	Read port 1 bit 0 again;
<0>/125	346 001		ANI	1	Mask all but bit 0 again;
<0>/127	202		ADD	D	Sum old bits with new bit;
<0>/130	017		RRC		Rotate new and old into next position;
<0>/131	127		MOV	D,A	Save result back in D;
*<0>/132	006 200		MVI	B,200	Time delay between bits;
<0>/134	005	WDATA	DCR	B	Decrement data wait count;
<0>/135	302 134 <0>		JNZ	WDATA	If not done then keep waiting;
<0>/140	035		DCR	E	Decrement data count loaded at 0/103;
<0>/141	302 123 <0>		JNZ	GETDATA	If not done then repeat for next bit;
<0>/144	162		MOV	M,D	Save received data in memory;
<0>/145	043		INX	H	Point to next available location;
<0>/146	174		MOV	A,H	Move high order address to A for end check;
√ <0>/147	376 yyy		CPI	yyy	Has high order address reached end?
<0>/151	302 103 <0>		JNZ	STARTBYT	If not then reiterate for next byte;
<0>/154	166		HLT		End input;

Notes:

- Input is assumed to be wired to bit 0 of port 1, from output of IC38 pin 6 via resistor R38 and shunted by diode DG1.
- Loading proceeds from split octal address xxx/xxx to address yyy/000. Enter this program by jumping to location <0>/100 after setting up constants of address.
- "*" indicates a timing constant for the "software UART" inputs.
- "√" indicates the end of transfer comparison mentioned in text.
- <0> indicates an arbitrary page location for this program, to be replaced by a real memory page number when actually loading the program at byte 100 of some page.

Listing 2: Stand Alone Suding Cassette Output Program. This program is a self contained data transfer routine which will transfer a block of data from split octal memory locations xxx/xxx through yyy/000 onto cassette tape after a five second leader output delay. This program assumes that CASTOMEM (see listing 1) will be used to read the tape being created. A more generally useful output facility would be modelled on this program and linked to a system monitor as a subroutine.

Split Octal Address	Octal Code	Label	Op.	Operand	Commentary
<0>/200	041 xxx xxx	MEMTOCAS	LXI	H,xxx/xxx	Load starting address in HL pair;
<0>/203	076 001		MVI	A,1	Start port output in high state;
<0>/205	323 001		OUT	1	Send initial state out;
<0>/207	026 012		MVI	D,012	Outer leader delay count;
<0>/211	006 377	LEADER5S	MVI	B,377	Outer leader delay loop return;
<0>/213	016 377	LEADER5X	MVI	C,377	Middle leader delay loop return;
<0>/215	015	LEADER5Y	DCR	C	Inner leader delay loop return;
<0>/216	302 215 <0>		JNZ	LEADER5Y	If inner loop not done then reiterate;
<0>/221	005		DCR	B	Middle leader delay count;
<0>/222	302 213 <0>		JNZ	LEADER5X	If middle loop not done then reiterate;
<0>/225	025		DCR	D	Outer leader delay count;
<0>/226	302 211 <0>		JNZ	LEADER5S	If outer loop not done then reiterate;
		*			
		* Upon reaching this point, 5 seconds of mark (high) state have			
		* been output to the cassette interface.			
		*			
<0>/231	016 011	BYTEOUT	MVI	C,011	Define output bit count (decimal 9);
<0>/233	257		XRA	A	Clear carry (start bit level is 0);
<0>/234	176		MOV	A,M	Move current byte to A;
<0>/235	027		RAL		Rotate bit into position (carry=0 first);
<0>/236	323 001	WNEXBIT	OUT	1	Send current LSB to output port;
*<0>/240	006 200		MVI	B,200	Time delay between bits;
<0>/242	005	WOUTLOOP	DCR	B	Decrement delay count;
<0>/243	302 242 <0>		JNZ	WOUTLOOP	If time left then reiterate;
<0>/246	037		RAR		Rotate new bit into position;
<0>/247	015		DCR	C	Decrement output bit count;
<0>/250	302 236 <0>		JNZ	WNEXBIT	If data left then reiterate;
<0>/253	076 001		MVI	A,001	Stop bit state defined
<0>/255	323 001		OUT	1	then sent out to port;
*<0>/257	006 377		MVI	B,377	Stop bit value set;
<0>/261	005	WIBDELAY	DCR	B	Decrement stop bit counter;
<0>/262	302 261 <0>		JNZ	WIBDELAY	If time left then reiterate;
<0>/265	043		INX	H	Increment memory address;
<0>/266	174		MOV	A,H	Move high order address to A for end check;
√ <0>/267	376 yyy		CPI	yyy	Has high order address reached end?
<0>/271	302 231 <0>		JNZ	BYTEOUT	If not then continue output process;
<0>/274	166		HLT		End output;

Note:

- Output is assumed to be wired from bit 0 of port 1 to DS1 in figure 2.
- See notes to listing 1 for listing conventions.

system flexibility. The "software UART" allows the timing constants to be dynamically modified (if desired) by detecting the variations in the stop bit timing, thereby compensating for wow and flutter. Digital integration of the incoming data bits is possible by setting a register to octal 200 at the beginning of each bit time. During the bit time, repeated sampling either adds or subtracts from the register (depending on whether 1 or 0) and a "branch minus" instruction system effectively eliminates receive problems. This digital integration detection is utilized by the Digital Group Z-80 cassette read software.

Versions of this "software UART" system have been written for 8008, 8080, Z-80, 6502, and 6800. All work satisfactorily.

Operation

This cassette system is utilized by first turning on the cassette recorder and waiting until the lower tone 5 second leader tone is heard. At this point, restart the system to the beginning address of the "Cassette to Memory" software.

Cassette writing is accomplished by restarting the system to the beginning of the "Memory to Cassette" programming. Be sure to set the appropriate start and stop addresses prior to beginning the read or write operations. The monitor programs in the various Digital Group systems automatically set the start and stop addresses. The check marks in the listing (√) indicate the points where start and stop addresses may have to be modified.

The software may be adjusted to run at different data rates by changing the values at the addresses mark with an asterisk (*). Note that the constants at <0>/133 and <0>/241 are the same. The constant at <0>/116 is 50% greater and the constant at <0>/260 is twice the value of the constant at <0>/241.

Summing It Up

This cassette interface represents a simple but fast and dependable way to store programs and data for the serious hobbyist. It does not seek to be all things to all users, but a number of applications can be run using the same basic design. The detail interface design has independence from other components in the system, allowing various processors to use the same cassette system (with appropriate software).■

Surplus Electronics in Tokyo and Manila

Dr Michael N Hayes
PO Box 167
Port Orchard WA 98366

Both Manila and Tokyo are characterized by a large number of very competitive shops which deal primarily in components.

Integrated circuits are a rarity in the surplus electronics markets. A shop in Manila keeps integrated circuits in a glass case similar to those found in a jewelry shop.

This article is a brief summary of some of my experiences, opinions, and observations while searching for and buying surplus electronic components in Tokyo, Japan, during October 1975; and in Manila, Republic of the Philippines, during November 1975. Perhaps this information will be of interest to US buyers and hobbyists for comparison purposes or in case they have the opportunity to visit these places.

In summary, the surplus electronics market in Tokyo is far better than that in Manila, but neither place offers the quantity of sophisticated computer electronics and integrated circuits available in the United States. The primary surplus electronic items sold in both cities are components: resistors, capacitors, switches, panel lights, terminals, connectors, transistors, etc. Component prices in Tokyo are roughly two thirds of US mainland prices, whereas in Manila the component prices are about the same as US prices. An important consideration here is that these components really aren't considered as surplus electronics in these cities, but are sold as retail electronics, subject to considerable bargaining, through hundreds of small shops and stands concentrated into specific areas of each city. Time did not permit investigation of industrial surplus outlets or auctions, but the impression formed is that the large numbers and competitiveness of these small stores preclude easy bargains from auctions or company sales.

Prices quoted in this article are of specific sampled items and are for illustration purposes only. All prices are in US dollars or cents with the following approximate exchange rates in effect: Tokyo, 300 yen per US $1; Manila, 7.5 pesos per US $1. There was a considerable disparity in prices among various stores; and thus, much like a marketplace, significant price savings could be found on selected items by spending the time to shop around. In many cases prices on selected items in one store, especially in Manila, might be half or double what was found in another store.

The real surplus electronics bargains of the Far East are rumored to be found in Taiwan, Hong Kong, and related areas. Perhaps a reader who has visited these areas could write in and supply additional findings.

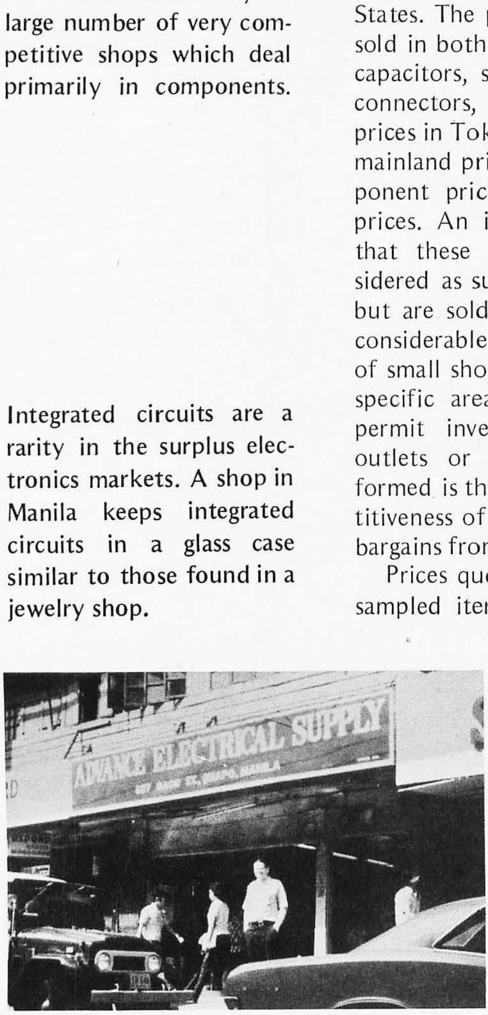


Photo 1: The author in front of a surplus outlet in Manila, Philippines.

Tokyo, Japan

The entire Tokyo surplus electronics market, along with significant portions of the retail electronics market, is concentrated in an area called Akihabara (pronounced ah-kee-ha-ba-rah), northeast of the downtown area. To reach it, one simply takes one of the many trains available from the elaborate Tokyo train and subway system.

What is incredible about Akihabara is not so much what they have, as the quantity in which they have it. There are literally hundreds of tiny outlets here selling everything imaginable in the way of electrical and electronic goods: televisions, stereos, speakers, tape recorders, radios, ham equipment, light bulbs, tools, wire, refrigerators, air conditioners, etc. Much of the electronic market is transistor radios and calculators, but the number of components stores and amount of available "surplus" electronic components to be found far exceeds any comparable area I have seen in any US city.

Each of these shops has components in counters, bins, or stands by the hundreds, with small signs marking the types and prices. Bargaining is expected in most stores but frowned upon in a few. Most of these shop owners are hard bargainers, but in most cases a 10 – 20 percent savings can result.

Here is a list of some sample items and prices in Tokyo:

- Ribbon cable, 10 conductor, Spectra Strip – 10¢ per foot.
- Toggle switches – 75¢ to $1.
- Small experimenter's PC cards (2 inch square) – 10¢.
- MC 7805s – $2.50.
- 44 pin PC board edge connectors – 75¢.
- Small capacitors – 5¢ to 10¢ .
- 7400s – 55¢ (there are not really too many ICs available).
- PC boards with components – 25¢ to 50¢ to $1 (most have Japanese numbered transistors and ICs).
- Resistors – 2¢ to 5¢.
- General 50 pin connectors – $2.
- Crystals – small types – $1.
- Partial calculator PC boards – no cases – older styles – $2.
- Partial cassette tape decks, audio, missing or damaged electronics – $1 to $3.
- IC sockets – 30¢ to 60¢.

Manila, Republic of the Philippines

The main area of Manila for surplus electronics is called Raon (pronounced rah-*own*) Street and is located in a district near the downtown area called Quiapo (pronounced Kee-*ah*-po). The name of Raon Street, including its signs, was recently changed to Gonzalo Puyat; but the Jeepney drivers still know it as Raon Street or Raon district. The best way to get to Raon is to take a Jeepney, which is any of the old WW II jeeps which have been painted and elaborately decorated and are now used as taxis throughout the Philippines. The ride will cost about 25¢ from anywhere in the downtown Manila area. English is spoken and used extensively in the Philippines making it very easy for an American to get around and to be understood.

The Raon District has about 30 small shops which sell electronic parts in display cases and parts bins. The setup is similar to that in Tokyo, but Manila has only a small fraction (less than 10 percent) of the number of stores. The quantity of merchandise available in each store was considerably less than in the typical Tokyo store. Bargaining in Manila is expected and a 20 to 30 percent savings is not hard to get.

I also found several stores in San Fernando, a small town about 40 miles west of Manila. These stores were very similar to the ones in Manila and seemed to support the local TV repair shops.

None of the shops in Manila or San Fernando had any computer equipment, and only a few of them had any integrated circuits. The integrated circuits were usually only a handful, mounted in styrofoam and kept by the owner in a glass display case, like jewels in a jewelry store.

Here is a list of some sample items and prices in Manila:

- Small capacitors – 10¢ to 20¢.
- 7400s – $1 to $3 depending upon the store.
- Alligator clips– 20¢.
- Panel lights, small – 50¢ to 60¢; no LEDs available.
- Medium sized capacitors – 20¢ to 30¢.
- Larger electrolytic capacitors – $1 to $1.50.
- Small variable capacitors – 75¢.
- Power transistors (TO3), general types – 20¢ to 30¢.
- Small potentiometers – 25¢ to 50¢.
- Regular metal potentiometers – 50¢.
- 12 Watt transformers – $1.50.
- Small plastic transistors –10¢ to 20¢.
- 7472 – $3, 7423 – $6, 7485 – $4, 7473 – $1.50, UA709 – $1.50 to $2, LM311 – $3.25.
- 44 pin PC board edge connectors – $1.40
- Transistor heat sinks – 75¢.
- Resistors – 5¢ to 10¢.

In conclusion, it's probably not worthwhile to make a special trip to either Tokyo or Manila to purchase surplus electronics from these markets. However, if one happens to be visiting either city, there are some selected bargains to be had by shopping around.■

It's probably not worthwhile to make a special trip to either Tokyo or Manila to buy surplus electronics; but if you happen to pass through either town on other business, the surplus markets are definitely worth a side trip.



Photo 2: Bargain hunting in Akihabara, a district of Tokyo.

The Great 8

With 8080 plug-in modules and software fron
the capabilities you've been dreaming of t



Sound Memory Judgment

Processor Technology now offers two options in low-power, fast static memory. Our read/write static memory modules—the **4KRA (4096 words)** and the **double-density 8KRA (8192 words)**—are **more reliable, less expensive, and require less power** than any other available modules of comparable capacity.

Both memories use the same **low-power** static RAM's (our prime supplier is AMD). All RAM's are manufactured to exacting military specification MIL STD-883C. The **2102A-4** RAM's used by our competitors are **not low power**—their data sheets show they use more than double the power (worst case) of ours. Both of our modules will retain data when powered by two "D" flashlight cells. (Connectors for battery power and re-charging are built in.)

The 8KRA has an exclusive address offset switch. This allows it to recognize any 8K address segment at 1K intervals, such as from 2K to 10K.

Each integrated circuit has its own premium grade, low profile IC socket (optional in the 4KRA). IC sockets make assembly, testing and repair many times easier. You'll quickly grow to appreciate them.

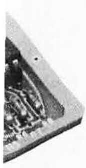
4KRA (without sockets or DIP switch)
4096-word Static Memory Module **$139**

4KRA (all sockets and switch incl) **$154**

8KRA 8192-word Static Memory Module
(all sockets included) **$295**

)80 Evolution

Processor Technology, you can now afford to add
›your Altair or IMSAI minicomputer system.



Accessible Software!

We firmly believe that good software should be easily obtainable for the hobbyist. You shouldn't have to resort to searching or copying source listings, just because some software can cost you as much as your entire minicomputer system. **Processor Technology now offers four well-tested software packages.** Their prices are within reach, and more are on the way!

Software Package #1

Resident Assembler/Editor/Monitor

SW#1 is very easy to use, even if you have little or no software background. It gives you power to develop and debug small to medium programs in 8080 assembly language. It allows program entry and editing by line number. Up to six program files may be stored at once in memory. This sort of Assembly Language programming should be used with any high speed, real time, or complex I/O applications. (Recommended total memory: 8192 bytes or more.) **Source Listing: $5.00—Paper Tape: $9.50.**

Software Package #2

BASIC

This conversational language is most useful in scientific, educational and recreational applications. Commands and statements are close to normal English. The floating-point calculations in our 5K BASIC are considerably faster than most other 8080 versions. (Any BASIC language is, however, best used in programs with fairly low speed requirements.) Ours is the only BASIC available that can store several programs in memory at once. Its formatted output provides for easier financial calculations, with multiple statements per line, and assembly language routine calling capability. (Recommended total memory: 8192 bytes or more, from address zero.) **Source Listing only: $5.00.**

FOCAL

(®Digital Equipment Corporation)

This is a math oriented language designed for scientific and educational applications. All programs written in standard DEC FOCAL for the PDP-8 will run on 8080 based machines, using our version of FOCAL. (Recommended total memory: 8192 bytes or more.) **Source Listing: $5.00 —Paper Tape: $9.50.**

Math Pack

Scientific Notation Calculator Program

Our Math Pack will provide basic math functions in a small memory system. In essence, it's a calculator that will add, subtract, multiply, divide and square root. You are allowed four variables, and you can set the decimal point (fixed or floating) between 0 and 12. Utilize either Algebraic or Reverse Polish logic—or mix them! (Program Requirements: 4096 total bytes, starting at address zero; 1300 bytes reserved for program space.) **Source Listing and Paper Tape (combination only): $9.50.**

Processor Technology manufactures a full line of plug-in modules compatible with the Altair and IMSAI minicomputers. **Write us now** for our free brochure or the location of your nearest PTCO dealer.

Processor Technology
6200-L Hollis Street
Emeryville, CA 94608

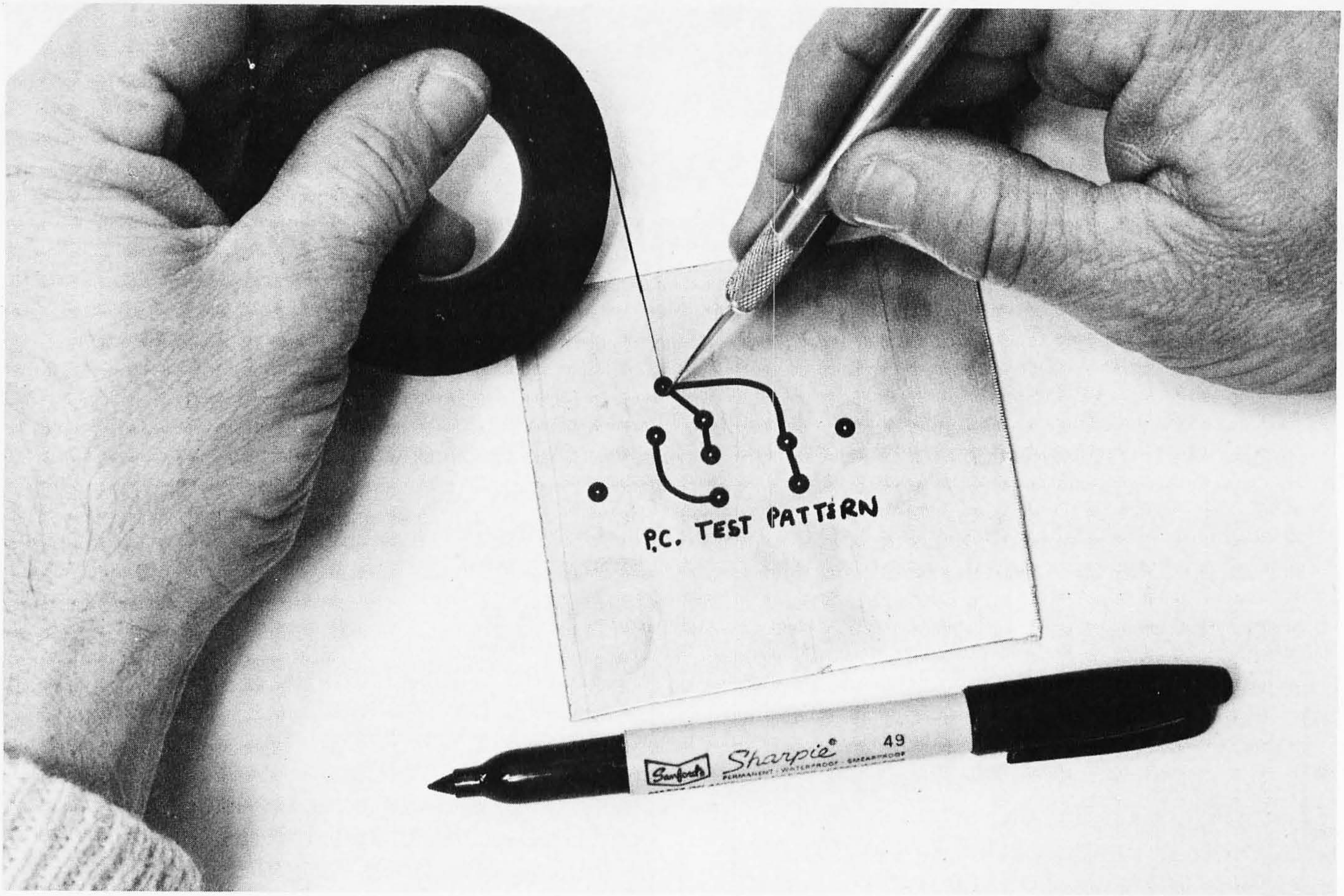


Photo 1: The Direct Etch Method. In this method, a one of a kind printed circuit is made by putting the pattern directly onto the copper. A Sanford's "Sharpie" pen (available in most stationery stores) can be used to draw patterns directly, and tape resist can be used for more uniform runs. If tape resist is used, care should be taken to avoid gaps in the adhesion of the tape to the copper.

Make Your Own Printed Circuits

James Hogenson
Box 295
Halsted MN 56548

Photos accompanying this article are by Ed Crabtree, using materials supplied by the author.

The widespread commercial use of printed circuits in electronic equipment began a few decades back when engineers started looking for more efficient wiring techniques to replace laborious hand-wiring methods. One of the first methods tried was to deposit (in other words, to print) a conductive ink pattern on a base of insulating material. The original method, printing, gave its name to all subsequent methods. Today, the term printed circuit refers to any electrical circuit in which individual wire lead connections have been replaced by a two dimensional conductive pattern bonded to an insulating base material.

Contemporary printed circuits consist of etched copper foil wiring patterns bonded to any of several insulating substrate materials sturdy enough to serve as a mounting base for the actual electrical components which make up the circuit. Although originally developed for mass production applications, printed circuit fabrication techniques have been refined until they can now be used by almost anyone with average mechanical skills.

Choosing your base material, the board, is a matter of price and purpose. The best is the epoxy glass board while phenolic (bakelite) is the cheapest. Phenolic base material is perfectly adequate for many applications, but since small boards are relatively inexpensive, epoxy glass is usually the optimum choice. The base material often comes

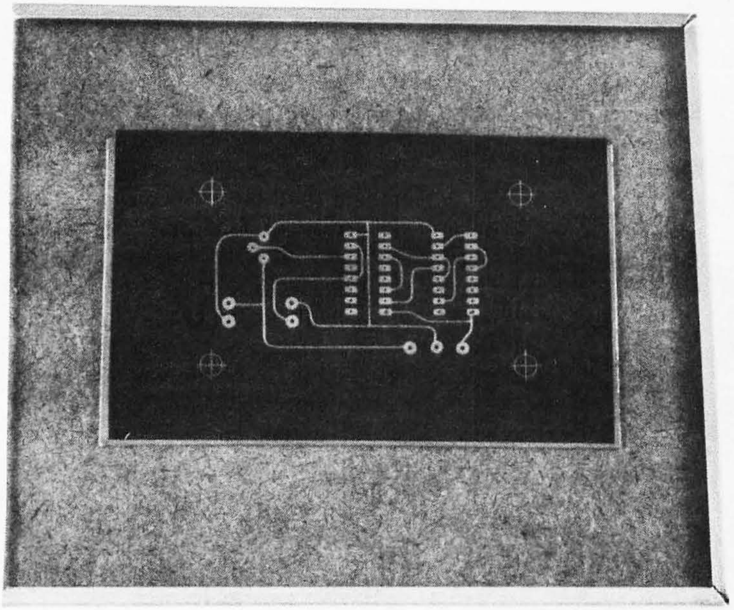


Photo 2: Printing the Circuit. Once a negative of the artwork has been created, the next step is to print the circuit. The negative is placed over a sensitized PC board and held firmly in place by a glass cover plate in the printing frame. The glass guarantees smooth and even contact for accurate transfer of the image. The board is then exposed to a photoflood lamp for one to three minutes.

laminated with copper foil on one or both sides.

The toughest part of making your first printed circuit board is getting started. In other words, the process may not be as difficult as you had thought.

A pattern of etch resist is applied by one of several methods to the copper foil. The board is then immersed in a chemical solution (usually a ferric chloride solution) which etches away all exposed copper. Then the board is washed and the etch resist pattern removed. The copper foil that was covered by etch resist remains on the board to provide you with a printed circuit.

Plan the Layout

The first step toward making your own printed circuit board is planning the layout. Draw the circuit pattern on paper as it should appear on the printed circuit board. You will use this as a guide for laying out the actual etch resist pattern. Keep in mind that you are looking at your board from the bottom when looking at the foil side. Be careful not to put the pattern on the printed circuit board upside down. (I've made that mistake more than once!)

Direct Etch

Direct resist is a method often used when a one of a kind board pattern is needed. Dry transfer etch resistant patterns are applied directly to the copper. The dry transfer patterns form integrated circuit pads, transistor pads, edge connectors, round donut pads, etc. Narrow etch resistant tape is applied to complete the circuit path between

A printed circuit is any electrical circuit in which individual wire leads have been replaced by a two dimensional conductive pattern bonded to an insulating base material.

The toughest part about making your own printed circuits is getting started.

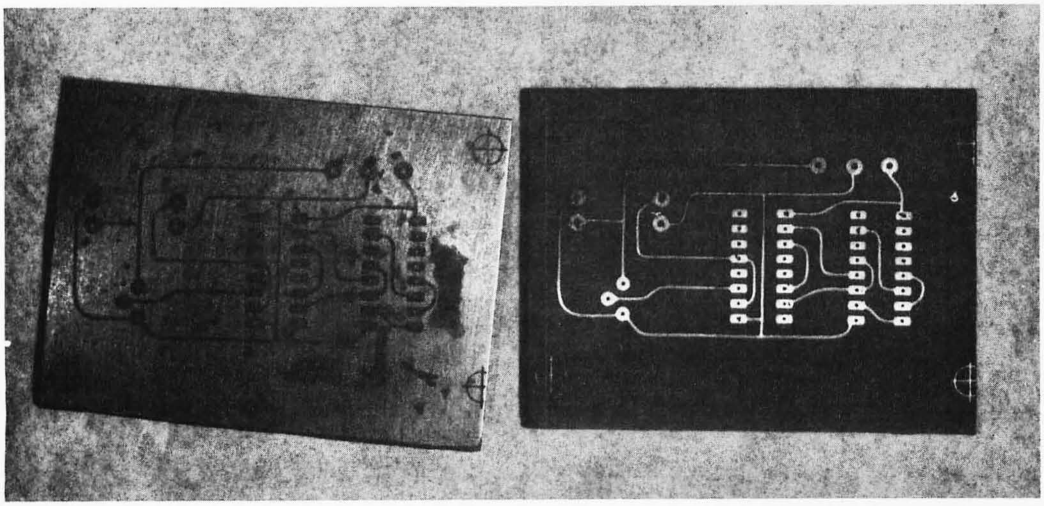


Photo 3: Results of Exposure. After being exposed, the photosensitive layer is developed, using an appropriate solution. An etch resist pattern will then remain on the board as in the example at left. (The dark blotches are oxidation on the copper.) The board is then etched with the usual ferric chloride solution. The finished product (hopefully free of imperfections) is a printed circuit board such as the one at right.

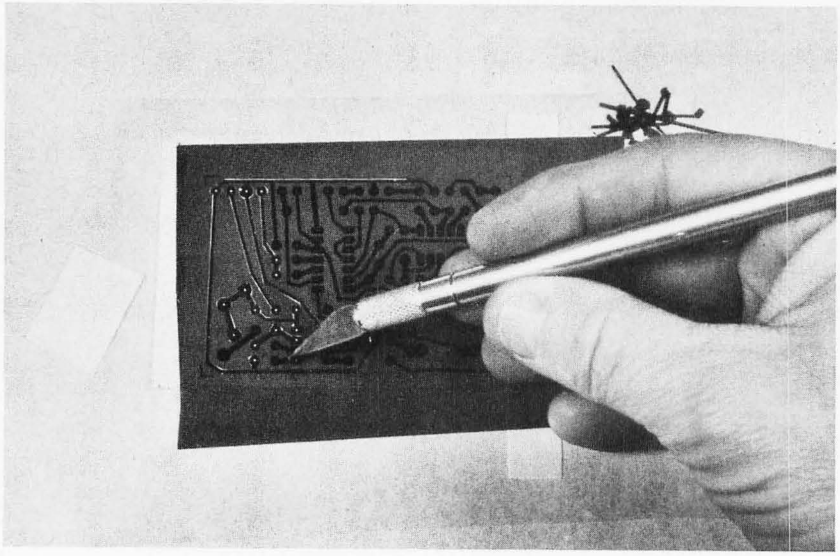


Photo 4: The "Cut-N-Peel" Method. A sheet of red mylar film on clear acetate backing is placed over the pattern to be copied. The negative is made by carefully tracing the pattern with a razor or sharp knife, then removing the red film wherever component pads and connections are to be made. (A trade name for the film used in this method is "Rubylith.") The negative is then transferred to sensitized copper and etched.

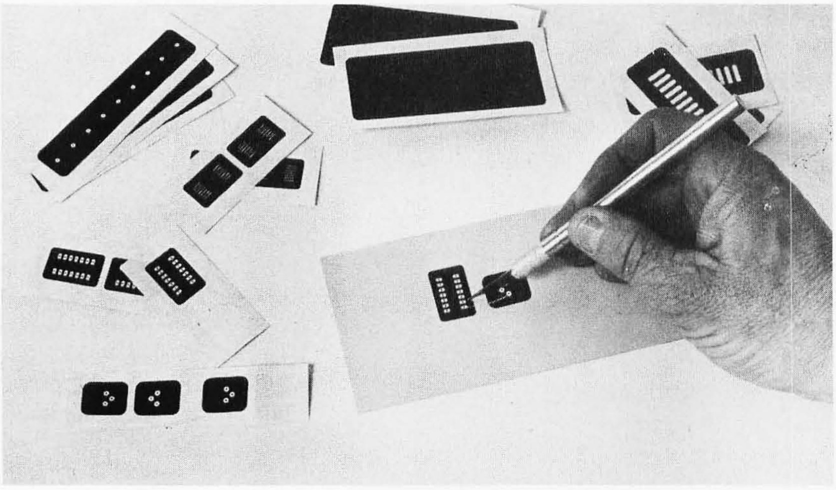


Photo 5: The Bishop Graphics " 'B' Neg" Method. In this method, a negative is made directly, using self adhesive black patterns on a mylar backing. The connections between patterns are made by cutting away the black layer with a sharp knife as in the "Cut-N-Peel" method.

component pads. Etch resistant ink pens and resist paint are also available for direct etching, as illustrated in photo 1.

After etching, copper will remain on the board only where dry transfer patterns or resist ink protected the copper foil from the etching solution. It should be noted that the etch resistant tape must be applied firmly, especially at overlaps, to keep the etching solution from getting under the tape and breaking the conductive copper path.

The direct resist method does not require extra steps for developing, as does the photo etch method. If only one printed circuit board is going to be made from a pattern, the direct etch method may be a time saver. If more than one board is to be made from one pattern, the direct etch method will quickly turn the element of time against you, since the pattern must be reconstructed on each board.

Photo Etching

The photo resist method is the most efficient method for making more than one printed circuit board of a kind. Photo resist etching is probably the most popular method, and is often preferred even for one of a kind printed circuit boards. The difference between photo resist and direct resist is the way the resist pattern is applied.

The copperclad board to be photo etched is first sprayed with a thin coat of a photo sensitive etch resist. This etch resist is sensitive to ultraviolet light. The sensitized board must be handled in a darkened room using a yellow light for illumination.

After the resist is dry, a negative of the printed circuit pattern is placed over the sensitized board in a print frame, as shown in photo 2. The board is exposed to the light of a photo flood lamp through the negative for one to three minutes. It is then immersed in a resist developer solution for about one minute. Only the etch resist which was exposed to the bright light will remain on the copper foil, as in photo 3. The resist is no longer light sensitive after developing, but should be allowed to dry for a short time. The board may then be etched. The copper which is protected by the remaining etch resist will not be removed. After the board has been etched, the resist is removed and the board may be cut, drilled, and assembled.

Making Negatives

It is plain to see that exposing a board through a reusable negative is much simpler than reconstructing the pattern by hand each time the pattern is used. The negative

may be obtained by a number of methods.

If a pattern is not too complex, the "Cut-N-Peel" method of photo 4 can be used. The pattern is simply cut into a red film on a clear acetate backing. The red film is peeled off, leaving a negative of the pattern.

If the pattern involves integrated circuits, the "Cut-N-Peel" method becomes rather difficult. The Bishop Graphics "'B' Neg" TM method would be more suitable. The ready made negative component patterns are laid out on a mylar sheet according to desired component placement. The areas between these self adhesive patterns are blacked out, using solid black acetate film. The only cutting necessary is for connections between component patterns. Photo 5 illustrates this method. The finished product is a negative of the entire printed circuit pattern.

Photographic Negatives

Perhaps the easiest and certainly the most popular method of obtaining the necessary negative is to first make a positive pattern, then produce a negative by photographic methods.

Positive artwork is made on a sheet of clear mylar film with matte finish on one side. This film is dimensionally stable and similar to plastic drafting film used by draftsmen. Positive artwork patterns are widely available in a large number of sizes and shapes. Photo 6 shows an example of a circuit being laid out with these patterns. Unless the artwork is going to be photographically reduced, use 1:1 artwork patterns. The self adhesive positive artwork patterns are laid out on the mylar sheet according to your pencil layout. Narrow black tape is used to form conductive paths between components. Graph paper or a similar grid should be used as a guide for orderly and uniform positioning of patterns. Since components are normally configured for dimensions which are multiples of 0.1 inch a 0.1 inch grid should be used.

A negative reproduction of your positive pattern can be made by a photographer or (preferably) by you. If you enjoy experimenting with photography, you might try experimenting with lithographic and orthographic films.

Photography Without a Darkroom?

The most popular negative producing method does not require photographic darkroom facilities. The special reversing film used may be handled in subdued light or in a darkened room using a dim yellow light. The positive pattern is placed directly on top

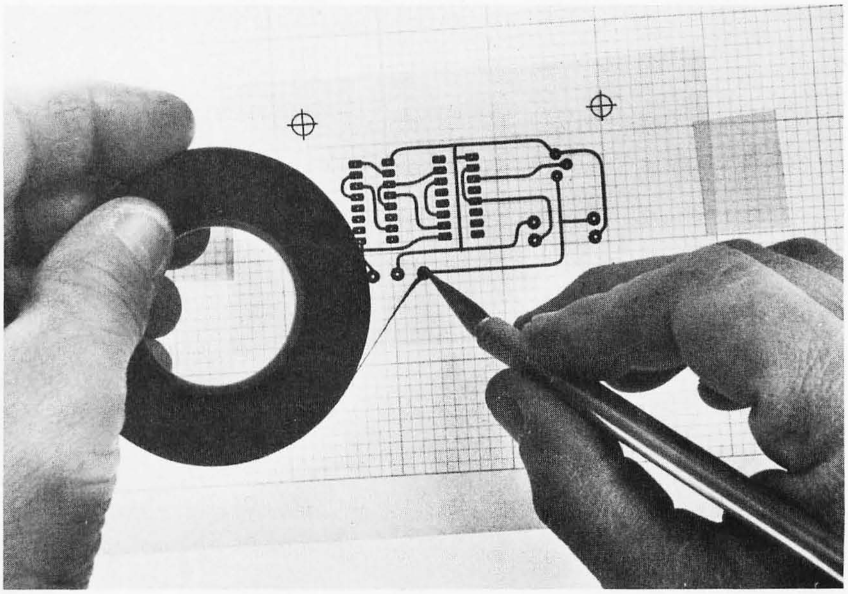


Photo 6: The Traditional Photo Negative Method. In this method, a positive artwork pattern is created, using preprinted self adhesive patterns and artwork tape. A sharp knife is used to cut the tape as it is being applied to the mylar film backing. A photographic process must be used to invert the image and create the negative form (see photo 5).

of the reversing film. The film is exposed through the positive artwork pattern to a photo flood lamp for one to three minutes and developed by rubbing gently with a cotton swab and a little film developing solution, as shown in photo 7. The opaque or colored emulsion on the film will rub off areas not exposed to light. The result is a clear pattern on a dark background.

A somewhat more involved but rather unique artwork developing system is made by Datak. With the Datak film and developing solutions, any of the following can be made: (1) negative from film positive or original artwork, (2) film positive from negative, (3) negative from negative, (4) film positive from film positive or original artwork, (5) film positive from black image on white paper, (6) film negative from black image on white paper. The last two methods allow you to copy a printed circuit pattern directly from a magazine page.

The Datak film is developed by methods similar to standard photographic procedures, so this method is more complicated and time consuming. Exposure and developing times are somewhat more critical. Datak film may, however, be handled in subdued tungsten light.

Advantages of Photo Resist Techniques

One of several advantages in using the photo resist method will become apparent when a modification of an existing board is

For a one shot printed circuit, simply draw the pattern onto copper with a resist pen and dump the board into ferric chloride until done.



Photo 7: Creating a Photo Negative for Etching. A negative is reproduced from the positive artwork pattern, using a reversing film. The film is exposed with a bright light, then developed by rubbing gently with a cotton swab and developing solution. The result is a negative version of the artwork with a 1:1 scaling.

For a unique approach to making jumpers on one layer boards, see Don Lancaster's "How to Build a Memory With One Layer Printed Circuits" in the April 1976 BYTE, page 28.

made. (Like when you need to make a board over because you forgot two or three connections. This **does** happen!) Rather than reconstructing an entire printed circuit pattern, make only the necessary changes or additions on the original artwork, then make a new negative and a new board. Making a new negative using reversing film requires only a few minutes of your time.

Double Sided Boards?

Sometimes a circuit will be too complex to fit on one side of a circuit board. Since a printed circuit is only two dimensional, conductor paths cannot cross. Jumper wires can be used to provide some crossovers, but if the circuit requires a large number of crossovers, a double sided circuit board might be considered. A double sided PC board is one which has a copper foil pattern on each side. The major consideration in making a double sided PC board is getting the pattern and terminals lined up. Both sides of the board are developed and etched at the same time.

Drilling

The step following the fabrication of a PC board is drilling out the holes. A small bench type drill press is ideal for this purpose. A standard hand held drill is unsatisfactory as the small drill bits break at low speeds. Commercially, small holes are drilled in boards at speeds as high as 70,000 RPM. A Dremel "Moto-tool" is a suitable compromise for work on printed circuit boards. This tool runs at 30,000 RPM. Such a tool will not only drill out extremely small holes, but cut and shape printed circuit boards, and lend itself to a host of other uses not related to making boards. A multipurpose tool like this is handy, especially for cutting out things like board edge connectors.

If repairs or small changes are needed on a printed circuit board, a piece of bare wire soldered over the foil is the cheapest and quickest modification. A conductive silver paint is available for printed circuit repairs, but the paint is quite expensive. GC Electronics, Techniques Inc, Kepro, and Datak each manufacture printed supplies for the

hobbyist in addition to their commercial products. Such supplies are distributed through a large number of mail order firms and retailers. The appendix lists the various products and who makes them. Cost of materials will vary depending upon a number of factors, but a figure of 20 cents per square inch of printed circuit board will provide a good rule of thumb to estimate the cost per board.

You will notice that Techniques and Kepro do not manufacture photo resist spray. Instead, they sell printed circuit board panels with the photo resist already applied. Presensitized panels (which come wrapped individually in dark paper) will assure you of a uniform and dustfree coating of photo resist. However, if you make a mistake developing the resist pattern, you will waste the extra cost of presensitized panels. It is a good idea to start with a spray resist, then graduate to presensitized panels once you have refined your circuit fabrication techniques. And the keys to refining your techniques are: Read instructions and familiarize yourself with what you're doing, follow the instructions, take your time, be careful, and practice first, using small sample boards. Follow those hints and you may surprise yourself with the fine boards you can turn out.■

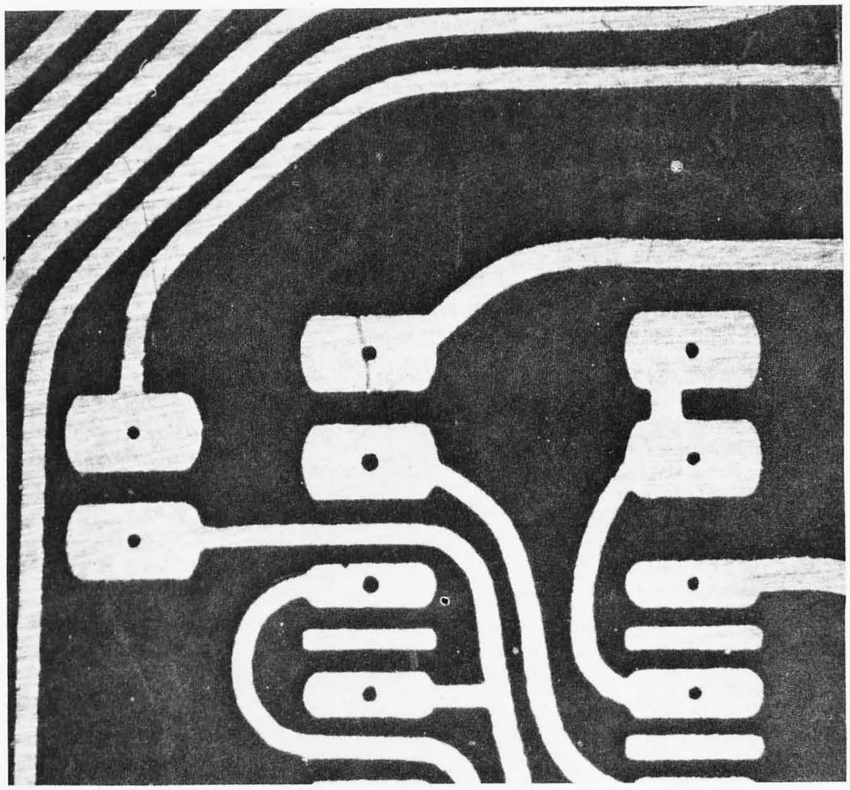


Photo 8: Close up, a successfully etched printed circuit will have even lines with no hairline cracks or other imperfections. This example shows such a result, prior to drilling out the holes for component leads.

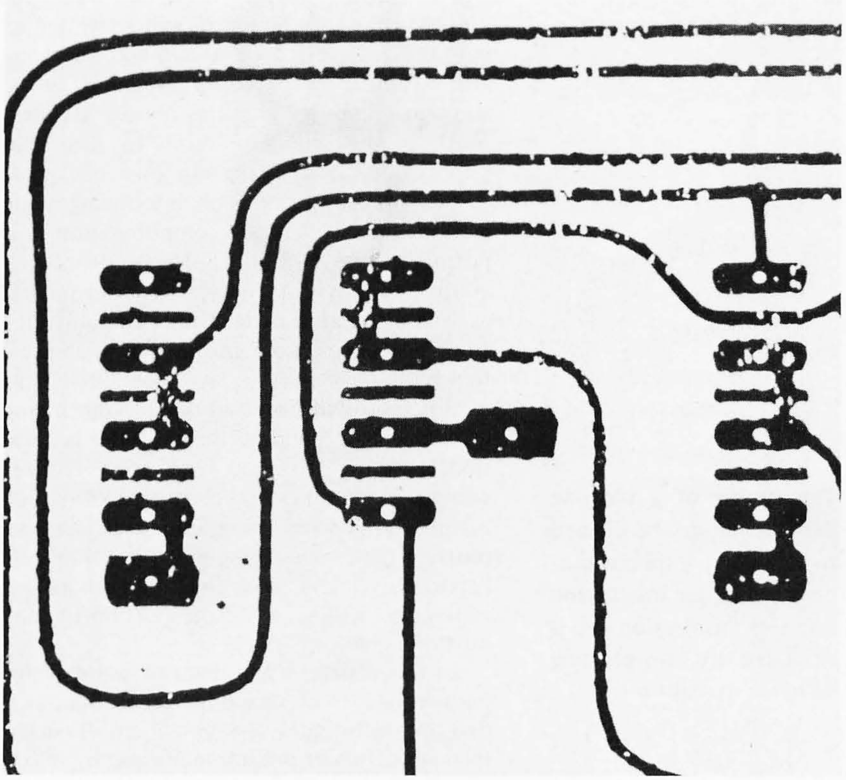


Photo 9: When various imperfections enter the picture, the result is not so clean. Here is a high contrast picture of an imperfect result. The resist layer has separated from the copper during the etch process at several points, resulting in holes in the copper and, in several instances, complete breaks in circuit runs.

APPENDIX: Sources of Supply

Direct etch materials

Ink resist is made by GC Electronics, Techniques, and Kepro.

Dry transfer resist patterns are made by Techniques, Datak, and Kepro.

An ordinary "Sanford's Sharpie" marking pen available for about 49 cents at any stationery store can be used as a resist pen.

Photo etch supplies

"Cut-N-Peel" and " 'B' Neg" supplies are distributed by GC electronics. (The " 'B' Neg" materials are manufactured by Bishop Graphics.)

Rubylith material, available at art supply houses, can also be used for cutting and peeling patterns.

Positive artwork patterns and supplies are made by Datak, Kepro, and Techniques. GC Electronics distributes artwork materials made by Bishop Graphics. Bishop Graphics materials are also distributed by independent distributors.

Photo etch supplies

Photo resist spray and developer are made by GC and Datak. Presensitized panels are distributed by Techniques and Kepro. Reversing film and developer are made by Techniques, Kepro, and Datak.

All of the above mentioned manufacturers make or distribute plain PC board panels (unsensitized) and etching solutions.

Photo flood lamps are available at photo supply houses. (Look for 375 Watt reflector flood lamp or No. 2 (EVB) Photoflood.)

A Plot Is Incomplete Without Characters

Richard J Lerseth
8245 Mediterranean Way
Sacramento CA 95826

Who would want to miss the opportunity of creating customized graphics for special applications?

As computer hobbyists, a number of us will sooner or later play around with graphics using vector CRTs or XY pen plotters; but very few of us will be willing to pay the high price of a number of copyrighted plotting packages available today through computer graphics houses. Besides, most of us will not want to miss the opportunity of creating our own packages.

So, in the process of interfacing your graphic media to your computer, you will normally have built the software needed to control simple vector moves on your media, as well as be able to window your plottings (that is, confine your moves within a specified area).

But, you will find that one of your major efforts will be building the character generation module. As you will soon realize, computer graphics take large chunks of memory space for the graphic routines and plotting tables describing plotting sequences. Particularly, you will find that character generation will take a large portion of that memory space.

The design of a plotting data format can be likened to designing a special purpose computer instruction set; this instruction set is emulated by the plotting software in real time.

In this article, I will describe some of the basic concepts of character generation, and describe techniques of saving memory space through efficient programming and by maximizing the packing of information in the plotting tables.

I assume at this point that you have within your basic plotting software: (1) the capability of shifting the relative origin within the plotting frame and (2) the capability of chain plotting. That is, plotting a vector from the ending point of the last vector move to the new position on the plotting field without explicitly defining the beginning point every time you make a vector move. We make full use of these two capabilities in plotting the character strings.

Plotting Frame

The easiest way of plotting a character is to define a plotting frame or grid upon which a sequence of vector moves are made from grid point to grid point. To minimize the complications involved with signed vectors, it is best to set the origin in the lower left hand corner of the field on which the character is to be plotted. With this convention, the vector moves are positive upward and to the right in the grid. In this way, we can define the ending point of a vector move with positive integer coordinates.

Limiting Frame and Plotting Resolution

Next, we have to define the resolution in plotting the characters. That is, we have to decide how many grid points we desire within a character frame. This depends on many factors: How fine you want your plotting; how many different characters you are to plot; how you are to pack the moves into memory; what special effects or options you desire. These considerations are all interrelated and must be considered as a whole.

I will propose an optimum character grid field within a limiting frame which will have a resolution as fine or finer than any used today by the graphic houses in their charac-

The choice of a character grid should reflect the realities of the common machine designs. For most microcomputers (and minicomputers), a character frame optimized for 8 or 16 bit words is desirable.

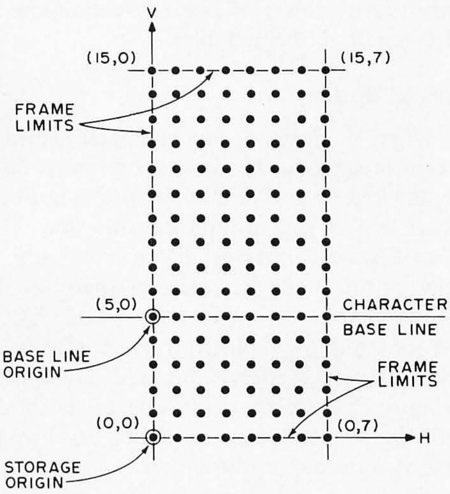


Figure 1: 8 by 16 Character Frame. Characters are plotted (or drawn on a vector graphics display) with reference to this local coordinate system. A series of 8 bit codes identifies the successive locations of the pen (or electron beam) and whether or not a line is to be drawn while moving to the location. The codes which reference the bottom row of this grid are treated as special operation codes for the plotting software: subchain reference, half shift right option, and floating subchain operation are defined in this article.

ter plotting packages. It will minimize the use of storage, and will also have some capability for special options. However, as a user of this software, you can make appropriate changes in your own system to reduce the resolution or to eliminate some of the special options.

Figure 1 shows the (8 x 16) grid I propose. The storage origin (0,0) is defined to be in the lower left hand corner of the grid. The character base origin (0,5) is at the lower third point of the left hand side of the limiting frame such that upper case alphabetic characters will be confined to the upper two-thirds of the grid frame. The lower third will be used for the tails of lower case alphabetic characters. The lower row of the grid will not be used for plotting; this row of 8 points will be reserved for flagging special options, which will be explained later.

Specifying Moves

Since most of us are using or will be using 8 or 16 bit machines, choosing this grid optimizes the packing of information for a vector move into an 8 bit byte of memory. A move to any point in this grid field (figure 1) could be defined with 3 bits for the horizontal (H) position, 4 bits for the vertical (V) position, and 1 bit for the Z function or the status (P) of the move (pen up or pen down for pen plotters, or intensity modulation in video graphics).

The 8 bits of H, V and P data for a move can be packed in the six different ways, such as HVP, VPH, PHV, VHP, HPV, or PVH. However, when packing such data into the byte, one must consider which is the fastest way to unpack the values. This greatly depends on the machine used. In most cases, it simply entails masking and shifting. I am going to use (VHP) as my standard. Why? No reason except that it can be implemented on most of the micros in use today without excessive effort. One procedure of unpacking the byte is given in appendix A.

To clarify further discussions on vector moves, the coordinates of the moves within the limiting frame will be written as (V,H). When the Z function is included, the move will be defined as (V,H,P) where

V is the vertical portion of the move
H is the horizontal portion of the move
P is the status of the Z function where,
 0 is pen down or display tube electron gun on
 1 is pen up or display tube electron gun off.

The lower portion of the grid (V = 0; H = 0 to 7; P = 0 or 1) will be reserved for special options which will be defined later.

Optimization of the Storage of Character Moves

Once a user starts playing around with developing the moves for each and every character, he soon realizes that there are a number of instances where a chain of moves is duplicated in the patterns of several characters. One can take advantage of this by building subchains and referencing them where it is appropriate to combine them in a large sequence. For example, the upper case

When implementing this software for a graphic display mechanism such as a CRT, pay attention to speed of execution. Flicker will result if your computer and software cannot keep up with your eye's timing characteristics.

alphabetic characters, (G, C, O, and Q) can all be combined together in one single chain. Also, the many lower case alphabetic characters have (c, ɔ, or o) as part of their chain. Taking advantage of such duplications can significantly lower the storage requirements of character plotting tables.

Special Options

When I defined the character limiting frame previously, I reserved the lowest horizontal line of grid points for special options. There are 8 grid points on this line. This gives 8 special options that can be used. If one considers the Z function, there are 16 options in all. Whenever (O,H,P) is encountered in a plotting chain of moves, then a special option is initiated. The special options can use the following bytes in the plotting sequence and, as such, can involve one, two, three, or more bytes.

With the character defined, the next task is to shift, twist, stretch or squeeze the characters as they are drawn.

The first special option we need is a subchain option. I shall define the code as (O,O,O). When this code is encountered, the next byte in sequence is the subchain number. As one can see, there can be 256 subchains. You will probably never need all 256 unless you build a large multi-language or multi-font character set.

The second option needed is a 1/2 shift right option. The code I used is (0,1,0). This option increases the resolution of the plotting in the horizontal direction and comes in handy when plotting upper case alphabetic M, T, V, W and a number of other characters to make them symmetric in the particular grid frame I propose. It is a one byte instruction to shift the horizontal portion of the next move byte one half grid space to the right. That is, if the sequence of bytes (5,0,0) (0,1,0) (5,3,0) (5,4,0) was encountered, then the next two moves would begin at (5,0), move to (5,3-1/2) and end at (5,4) with pen down (or gun on).

These last two options I consider to be the minimum you should have in your system if you are to have the resolution required to plot large character sets.

Another option that could be used is the floating subchain option, (0,2,0). (This option is not shown in figure 6.) It takes three bytes of code to complete the sequence of this option. For instance, a period is used extensively for a number of punctuation characters and lower case i's and j's. The subchain sequence (1,0,1) (1,1,0) (2,1,0) (2,0,0) (1,0,0) plots a period in the lower left hand corner of the grid. Now, by using the floating subchain option, this period can be floated anywhere on the grid. A three-byte sequence (0,2,0), (SV,SH,0), (subchain no.) will move the period to any location desired by using positive offset values (SV,SH). This would save at least two bytes of storage for every different position of the period in the grid field, if there are more than two positions to be plotted. But, it takes some extensive programming to include this option; the advantage is large in large character sets, but minimal in small sets. Also, since timing is important in using CRT graphic systems, one must consider whether the extra computing effort is worth the savings in memory. I will leave it up to you to dream up exotic plotting options of your own for the 13 additional options which remain undefined.

Pointer and Move Sequence Tables

The pointer and move sequence tables now have to be established. A general schematic of the tables is shown in figure 2, along with the relationship of the tables to one another.

The primary pointer table defines the starting point in the character vector move sequence table, and the number of moves for each particular sequence. The pointer table is two bytes per character and shown in figure 3. Five bits of the first byte gives a maximum number of 31 primary steps per character. This is more than enough for any character contemplated, even if it were script or gothic. It is conceivable that a sequence table can be as large as 8 pages or 2 K bytes long. The remaining 3 bits of the first byte could designate the page number.

APPENDIX A
UNPACKING A VECTOR MOVE FROM AN EIGHT BIT BYTE

Using V, H and P to denote bits, the move is VVVVHHHP in packed form. The unpacking procedure is as follows:

1. An arithmetic shift right will make the Z function of the move available in the carry flag. The user can make use of this information through appropriate compares and jumps. Note that masking all but Bit P will also make the Z function available, but the action of shifting also readies the horizontal position of the move.
2. Temporarily store present value of the accumulator in any other register.
3. Mask the accumulator with octal 7. The horizontal position is now available. Send it out to the graphic device or store it for later use in another register.
4. Bring back the stored value of the accumulator from Step 2, shift right three times and mask the result with octal 17. Now the vertical portion is available.

The 8008 microprocessor assembly code would look like:

032	RAR	Shift right.
310	LBA	Load results temporarily in Register B.
}	User defined portion using the Z function code in the carry flag.	
301	LAB	Load ACC with value in Register B.
044 007	NDI 007	Mask the ACC with 007_8.
}	User defined portion using the unpacked horizontal portion of the move.	
301	LAB	Load ACC with value in Register B and rotate right three times.
012	RRC	
012	RRC	
012	RRC	
044 017	NDI 017	Mask ACC with 017_8.
}	User defined portion using the unpacked vertical portion of the move.	

The second byte would designate the starting point within that page.

This two byte table will fit into one 256 byte page of memory if there are 128 or less characters in your set. So, the full ASCII character set would fit easily in one page. The 7 bit ASCII code, if it resides in the upper portion of the address byte (bits 7-1) with a zero in the LSB of the byte, can address the location table directly. The location table for the subchains will also use the same format.

In figure 4, I give my version of the full ASCII 128 character set. Tables 1-3 give the values needed to plot this set. The tables contain octal 2235 (decimal 1181) bytes of data. The tables are set up so that you can easily reduce the size of the tables to a minimum set containing only 63 upper case alphabetic, numeric, and punctuation char-

Table 1. **PRIMARY POINTER VALUES**

Octal Address	Octal ASCII Code	Decimal Starting Location	Decimal No. of Moves	Octal 2-Byte Packed Code (see figure 3)
000	000	587	11	132-113
002	001	598	17	212-126
004	002	615	12	142-147
006	003	627	8	102-163
010	004	635	10	122-173
012	005	645	12	142-205
014	006	657	14	162-221
016	007	671	9	112-237
020	010	680	10	122-250
022	011	690	8	102-262
024	012	698	8	102-272
026	013	706	6	062-302
030	014	712	7	072-310
032	015	719	11	132-317
034	016	730	2	022-332
036	017	732	4	042-334
040	020	736	11	132-340
042	021	747	7	072-353
044	022	754	9	112-362
046	023	768	8	103-000
050	024	776	6	063-010
052	025	782	10	123-016
054	026	792	14	163-030
056	027	806	14	163-046
060	030	820	6	063-064
062	031	826	8	103-072
064	032	834	6	063-102
066	033	840	10	123-110
070	034	850	4	043-122
072	035	854	6	063-126
074	036	860	9	113-134
076	037	869	6	063-145
100	040	875	1	013-153
102	041	245	11	130-365
104	042	262	12	141-006
106	043	274	8	101-022
110	044	278	14	161-026
112	045	292	9	111-044
114	046	301	12	141-055
116	047	256	6	061-000
120	050	323	6	061-103
122	051	329	6	061-111
124	052	313	6	061-071
126	053	317	6	061-075
130	054	243	7	070-363
132	055	317	2	021-075
134	056	245	5	050-365
136	057	335	2	021-117
140	060	147	13	150-223

Table 1 (continued):

Octal Address	Octal ASCII Code	Decimal Starting Location	Decimal No. of Moves	Octal 2-Byte Packed Code
142	061	160	5	050-240
144	062	165	9	110-245
146	063	174	11	130-256
150	064	185	4	040-271
152	065	189	9	110-275
154	066	206	11	130-316
156	067	217	5	050-331
160	070	198	17	210-306
162	071	222	11	130-336
164	072	233	10	120-351
166	073	238	12	140-356
170	074	344	3	031-130
172	075	337	4	041-121
174	076	341	3	031-125
176	077	347	12	141-133
200	100	359	19	231-147
202	101	0	8	100-000
204	102	8	12	140-010
206	103	23	8	100-027
210	104	34	7	070-042
212	105	41	6	060-051
214	106	41	5	050-051
216	107	20	11	130-024
220	110	47	6	060-057
222	111	53	8	100-065
224	112	61	6	060-075
226	113	67	6	060-103
230	114	73	3	030-111
232	115	76	6	060-114
234	116	82	4	040-122
236	117	23	9	110-027
240	120	86	7	070-126
242	121	23	11	130-027
244	122	86	9	110-126
246	123	95	12	140-137
250	124	107	6	060-153
252	125	113	6	060-161
254	126	119	4	040-167
256	127	123	6	060-173
260	130	129	4	040-201
262	131	133	8	100-205
264	132	141	6	060-215
266	133	378	4	041-172
270	134	382	2	021-176
272	135	384	4	041-200
274	136	388	3	031-204
276	137	391	2	021-207
300	140	393	2	021-211
302	141	395	10	121-213
304	142	407	10	121-227
306	143	397	8	101-215
310	144	418	4	041-242
312	145	397	10	121-215
314	146	422	8	101-246
316	147	430	7	071-256
320	150	437	7	071-265
322	151	444	10	121-274
324	152	446	10	121-276
326	153	456	6	061-310
330	154	462	5	051-316
332	155	467	12	141-323
334	156	479	4	041-337
336	157	409	9	111-231
340	160	483	4	041-343
342	161	487	4	041-347
344	162	491	6	061-353
346	163	497	12	141-361
350	164	512	6	062-000
352	165	518	7	072-006
354	166	525	4	042-015
356	167	529	5	052-021
360	170	534	4	042-026
362	171	538	7	072-032
364	172	545	6	062-041
366	173	551	7	072-047
370	174	565	2	022-065
372	175	558	7	072-056
374	176	567	4	042-067
376	177	571	16	202-073

acters. Appendix B explains how to reduce the size of the tables to the minimum set. But, I encourage you to go in the opposite direction and build up other subsets to add to this basic set. For example: Greek alphabet and mathematical sets, or centered symbol sets for line graphs.

Table 2. **SUBCHAIN POINTER VALUES**

Octal Address	Decimal Subchain Code	Decimal Starting Location	Decimal No. of Moves	Octal 2-Byte Packed Code
000	1	397	8	101-215
002	2	409	8	101-231
004	3	438	6	061-266
006	4	598	6	062-126
010	5	617	10	122-151
012	6	627	6	062-163
014	7	604	5	052-134
016	8	578	4	042-102
020	9	578	9	112-102
022	10	671	5	052-237
024	11	617	6	062-151
026	12	701	5	052-275
030	13	598	11	132-126
032	14	571	7	072-073
034	15	571	11	132-073
036	16	587	4	042-113
040	17	665	6	062-231
042	18	629	4	042-165
044	19	784	6	063-020
046	20	802	4	043-042
050	21	591	4	042-117
052	22	810	10	123-052
054	23	810	4	043-052
056	24	627	5	052-163
060	25	842	6	063-112

Position, Orientation, and Scale

Now that we have the ability to pull out the coordinates for a sequence of moves, we have just begun the job of plotting a character chain. We must translate each character into its appropriate position on the plotting media, then scale it up or down, rotate it into the proper position, and if desired, slant the character. What usually is done is to build conversion coefficients prior to plotting the desired character string. While going through the process of plotting, these coefficients transform the move coordinates residing in the move sequence table to the appropriate coordinates on the plotting media.

This requires that you have the capability of multiplying and dividing floating point numbers in your system. I assume you will either have a calculator chip interface or a floating point software package to draw on. Additionally, you will need the capability of obtaining sines and cosines if you want the ability to rotate the character string out of a horizontal position or to define the slant of a character with an angle.

Before we get into the procedure of shifting, twisting, stretching or squeezing the characters onto the plotting media, we must define a few parameters which are required prior to plotting the character string. In

Table 3.
MOVE SEQUENCE VALUES

000	040	100	140	200	240	300	340	000	040	100	140	200	240	300	340
121	167	132	124	376	325	176	224	307	234	347	226	365	242	346	262
320	076	176	132	121	370	256	260	350	174	002	206	370	142	326	000
364	121	376	176	376	130	312	320	370	152	146	147	130	355	330	003
372	360	121	216	361	125	300	364	366	142	371	144	124	134	271	263
336	372	360	252	136	134	360	372	346	365	346	124	241	000	266	022
136	336	377	244	361	321	374	336	350	320	264	126	304	001	126	000
221	176	200	300	002	364	273	276	305	360	224	146	236	353	026	002
236	132	245	320	226	372	314	172	346	376	146	273	101	372	062	000
121	120	136	364	376	336	354	124	366	120	130	310	116	364	363	001
360	241	361	372	002	276	372	311	364	137	367	306	323	124	122	275
372	250	120	336	227	254	364	330	344	132	350	264	214	127	143	034
354	377	136	361	002	202	342	326	346	176	272	224	275	122	274	263
314	360	121	376	126	120	302	306	311	136	232	206	134	243	207	264
272	120	360	002	213	136	264	310	352	255	150	210	155	250	134	124
260	136	002	367	304	363	221	171	372	130	126	232	132	000	371	205
273	121	126	002	361	376	264	210	370	124	125	313	124	001	366	272
236	360	376	126	376	314	272	206	350	160	372	212	142	275	126	234
176	241	136	361	120	272	236	166	352	220	301	174	242	054	125	143
132	256	121	160	136	266	176	170	221	326	316	216	264	032	130	124
120	377	360	124	125	273	132	067	236	346	201	336	272	024	121	132
251	136	136	132	142	236	124	130	277	364	216	372	254	042	260	154
256	365	376	176	200	176	160	131	260	362	161	364	214	363	241	174
136	372	121	376	300	132	220	150	367	340	256	320	202	122	262	212
337	002	360	361	342	124	324	146	124	320	320	160	363	243	264	204
372	367	372	002	364	160	372	126	131	136	337	124	122	264	246	222
364	002	336	126	372	133	321	130	372	143	240	132	143	272	126	242
320	126	276	376	354	372	360	002	355	354	176	373	124	254	247	264
160	125	232	361	316	160	376	167	344	343	341	366	132	134	270	272
124	132	220	122	216	176	210	370	322	154	364	126	154	125	272	254
132	221	231	002	154	161	130	366	262	241	372	132	254	130	254	000
176	160	136	246	132	124	277	002	244	256	336	365	272	331	134	000
336	124	161	134	124	132	232	166	252	002	254	132	264	350	123	000
037	**077**	**137**	**177**	**237**	**277**	**337**	**377**	**037**	**077**	**137**	**177**	**237**	**277**	**337**	**377**

UPPER CASE ALPHA — NUMERALS — PUNCTUATION — LOWER CASE ALPHA

An Aside:

The techniques used in this article can be directly applied to any repeatable set of plotting sequences for display on a vector graphics device. For example, the chess pieces and chess board of a chess game display are one possible data display; similarly, a Space War game's space ship symbol output to a graphic display device could use techniques of vector generation and rotation.

figure 5, we see that we need the standard height (S) and width (W) of each character, the gap (G) between each character, the starting coordinate position (X_O,Y_O) of the character string defined as the baseline origin (identical to the relative origin), and the angles (θ&β) defining the orientation and slant of the character string. These parameters must be made available prior to plotting the character string.

Now, let's list the formulae you will use in your plotting routine.

1. Scale Equations
   - $SS = S/10.0$ vertical scale (1)
   - $SW = W/7.0$ horizontal scale (2)
   - $SG = G/SW$ width-gap ratio (3)
2. Rotation Equations
   - a. Horizontal (H) portion of move
     - $HX = \cos\theta$ (4)
     - $HY = \sin\theta$ (5)
   - b. Vertical (V) portion of move
     - $VX = -\sin\theta = -HY$ (6)
     - $VY = \cos\theta = HX$ (7)
   - c. Vertical (V) portion of move corrected for the slant
     - $VX = -HY + HX*\sin\beta$ (8)
     - $VY = HX + HY*\sin\beta$ (9)
3. Final Coefficients for Rotation and Scale
   - $DHX = HX*SW$ (10)
   - $DHY = HY*SW$ (11)

000	040	100	140	200	240	300	340	000	040	100	140
263	274	264	264	176	346	225	000	000	002	272	320
272	205	260	171	002	324	230	016	017	166	172	325
331	212	273	070	172	306	361	265	171	071	000	266
346	263	264	131	002	266	002	164	176	170	004	000
126	274	164	136	072	000	262	172	076	076	000	031
132	122	172	177	000	006	366	177	070	176	025	361
263	134	225	076	006	000	000	170	133	000	000	260
142	373	230	000	165	011	013	070	136	006	026	266
124	366	171	004	264	000	367	076	000	000	000	366
132	270	070	265	172	012	360	131	017	013	006	000
154	246	076	272	272	000	260	134	117	177	273	031
275	230	261	002	115	006	321	000	110	170	264	137
134	126	360	267	076	273	324	017	174	070	224	
263	132	266	002	070	264	000	153	074	076	232	
002	365	366	166	170	224	014	174	000	116	172	
126	370	265	171	176	232	367	074	020	134	164	
274	266	164	076	076	172	360	073	165	130	000	
261	250	172	071	261	164	260	076	264	135	027	
122	226	272	176	360	261	266	000	272	156	000	
206	130	171	321	366	360	165	017	172	176	030	
132	124	070	324	266	321	264	151	225	000	000	
274	371	076	367	321	326	272	172	232	022	031	
123	130	367	360	326	367	232	176	000	000	000	
274	221	360	260	000	266	224	136	021	023	022	
263	264	320	266	010	000	231	110	000	000	326	
134	232	326	000	171	013	172	070	004	024	324	
263	276	266	005	070	361	000	076	265	000	000	
124	261	260	000	111	260	015	000	224	006	031	
132	360	265	006	176	266	000	000	232	165	261	
025	364	164	000	133	273	004	000	272	264	360	
030	346	172	007	076	264	271	000	002	002	366	
132	306	272	171	367	164	170	000	227	226	326	
037	**077**	**137**	**177**	**237**	**277**	**337**	**377**	**037**	**077**	**137**	**177**

LOWER CASE ALPHA | **ADDITIONAL PUNCTUATION** | **ASCII CONTROL CHARACTERS**

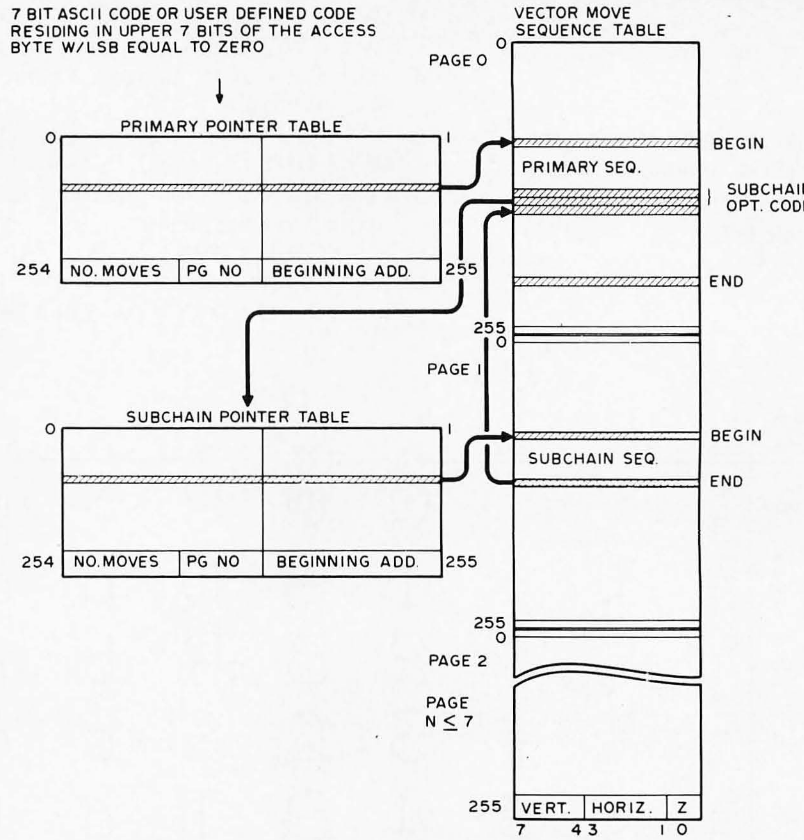


Figure 2: Relationship of the Character Generation Tables. The selected character code is rotated left by one bit to define a number from 0 to 254. This number accesses a 16 bit quantity in the primary pointer table. ***The primary pointer table in turn locates the beginning of a series of pen locations in the move sequence table which define the character's plot representation. Within that series, there might be a pointer to the subchain table, which in turn points to an often used fragment of the graphics representation located at a different place in the move sequence table. Note that to minimize retrieval effort on machines such as the 8008 and 8080, sequences of moves should be restricted to single pages of memory.***

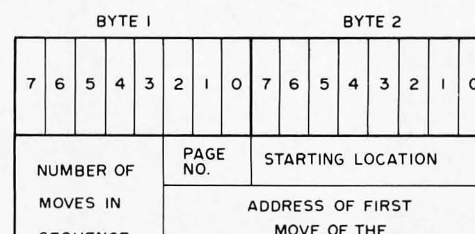


Figure 3: Primary and Subchain Pointer Formats. The pointer tables are composed of two byte elements which contain information on the number of moves required, and the address of the first move of the sequence.

As always, climb the highest mountain rather than be content with a mole hill.

$$DVX = VX*SS \quad (12)$$

$$DVY = VY*SS \quad (13)$$

4. Shift Coefficients Between Character Baseline Origins

$$DSX = DHX\ (7.0 + SG) \quad (14)$$

$$DSY = DHY\ (7.0 + SG) \quad (15)$$

5. Final Transformation Equations to be Applied to Each Move

$$X = XO + H * DVX + (V-5.0) * DHX \quad (16)$$

$$Y = YO + H * DVY + (V-5.0) * DHY \quad (17)$$

6. Shift Relative Origin from Character

$$XO = XO + DSX \quad (18)$$

$$YO = YO + DSY \quad (19)$$



Figure 4: an ASCII Graphic Character Set. The plotting tables 1-3 are used to define this set of characters when displayed or drawn on an XY plotter. This figure was prepared by the author, using a commercial plotter as the output device.

Formulas (1) through (15) are calculated prior to plotting the first character of a line. The coefficients thus derived will not change throughout the plotting of the character chain. Note that the equations simplify considerably when the angles θ and β are limited to special cases. Two common special cases are $\theta = 0°$, $\beta = 0°$ and $\theta = 90°$, $\beta = 0°$. Substituting the special values of sine and cosine for these angles produces the special cases. These values are:

$$SIN(0) = 0.0 \quad COS(0) = 1.0$$

$$SIN(90) = 1.0 \quad COS(90) = 0.0$$

Equations (16) and (17) are the transformation equations used during the plotting where only the values (H) and (V) change for each move. XO and YO are updated as we move to the next character in line to be plotted by using equations (18) and (19).

Plotting Routine

The plotting routine is outlined in figures 6 and 7 as a flow chart. If you have BASIC, you should not have any problems imple-

APPENDIX
ABRIDGING THE ASCII PLOTTING TABLES

NOTES:

1. To abridge the plotting tables, do the following:
   A. For upper case alphabetic, numerals, and punctuation, only use:
   Primary Pointer Table — bytes (octal) 100 to 301
   Subchain Pointer Table — none
   Move Sequence Table - bytes (octal) 0 to 613
   B. For all characters except ASCII control characters, use:
   Primary Pointer Table — bytes (octal) 100 to 375.
   Subchain Pointer Table — bytes (octal) 0 to 5
   Move Sequence Table — bytes (octal) 0 to 1070
2. If you want abridged Set A above, note that you do not need to include the traps for special subchain option in your program.
3. Note that the move sequence table is set up so that no sequence of moves crosses the boundary of a 256 byte page of memory. This eases the programming of micros such as the Intel 8008 or 8080.
4. Note that the blank or space character was included at the end of the move sequence table. If you abridge the table, move the code to the end of your abridged table and correct the location code in the primary pointer table. Better yet, include in your program a trap to catch any spacing, as there is no actual plotting for this character. Just shift the relative origin to the next character to be plotted.

menting this routine, as BASIC has the required floating point arithmetic and the transcendental functions, sine and cosine. If you plan to implement the routine in machine language, then I dare say you will have a little more work cut out for you. But, the advantage of going this route is that you will take full advantage of your microcomputer's design in order to minimize the use of memory and increase the speed of plotting. Speed is very important if you have a CRT graphics terminal, because of the refreshing problem.

Summary

In summary, I think you have here a start in creating your own vector character generation package on whatever graphic media you have or plan to use. You can implement the package as I have given it to you or abbreviate, expand, or abridge the package to suit your needs.

I encourage you, though, to expand the

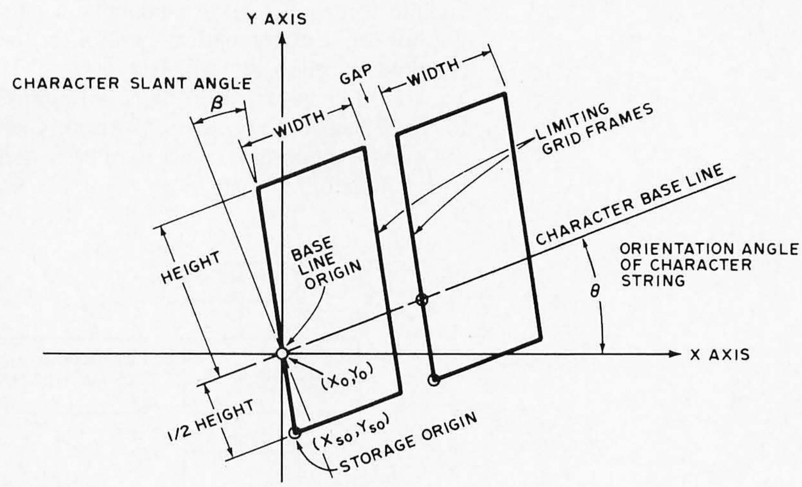


Figure 5: Character Orientation. To add an element of finesse to the plotting function, provision for general purpose rotation and slanting is a desirable feature. There are two angles to specify: angle θ is the orientation angle of the baseline for a character string; angle β is the frame slant relative to a perpendicular through the base line.

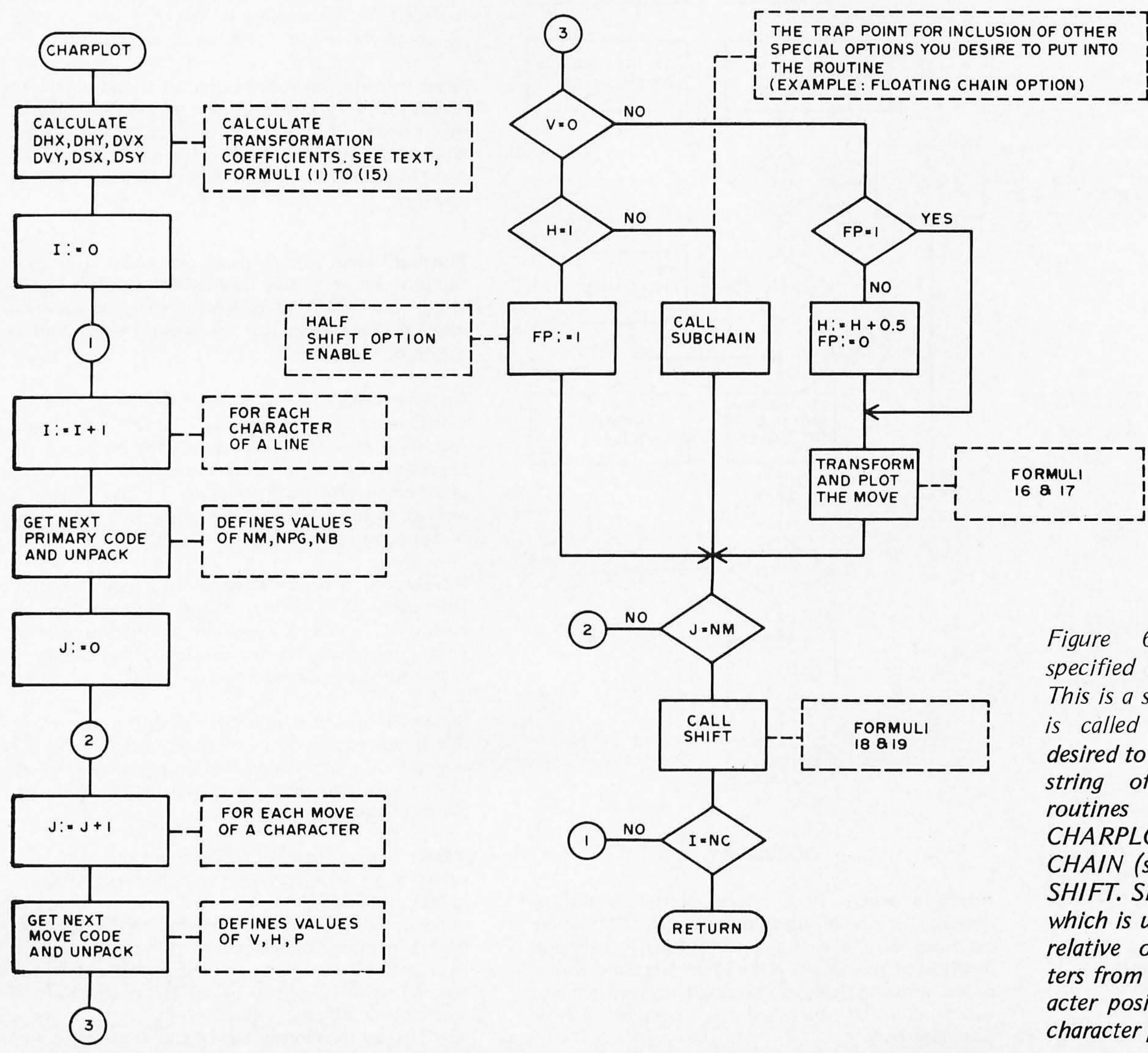


Figure 6: CHARPLOT specified as a flow chart. This is a subroutine which is called whenever it is desired to draw a character string of output. Subroutines referenced by CHARPLOT are: SUBCHAIN (see figure 7) and SHIFT. SHIFT is a routine which is used to move the relative origin of characters from the present character position to the next character position.

basic character set I have given you to include foreign language alphabets, a music symbol set, a mathematical symbol set, or a centered symbol set for line graphs. The horizon in character plotting is only limited to your own efforts or imagination. Climb the highest mountain, rather than be content with a mole hill.■

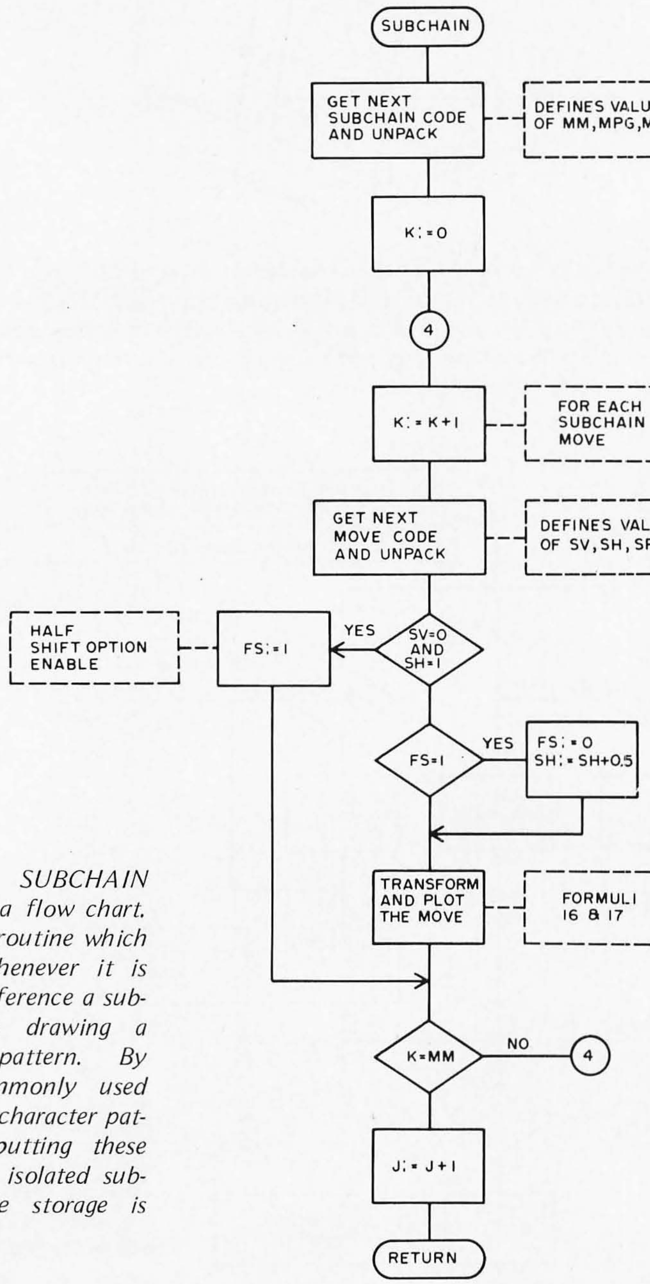


Figure 7: SUBCHAIN specified as a flow chart. This is a subroutine which is called whenever it is desired to reference a subchain when drawing a character pattern. By picking commonly used segments of character patterns and putting these segments in isolated subchains, table storage is conserved.

GLOSSARY

Absolute origin: In a typical plotter or display device, there is an absolute origin for all possible positions of the writing mechanism. A common location of this origin is the lower left hand corner of the plotting field, so that points to the right and above can be specified by positive integer displacements.

Byte: A cell in memory which can store 8 bits of information.

Chain: A set of vector moves to be performed sequentially.

Chain plotting: The technique of specifying a movement of the plotting or display mechanism by a series of small movements.

Character frame: A small region of the plotting medium in which motions will take place while plotting a single character. See figure 1.

Coordinates: A point in a two dimensional space can be specified by a pair of numbers. These numbers are the coordinates of the point with respect to a reference point called the origin.

Masking: The technique of selecting bits for inspection using the AND operation and a mask. The word which is being tested is combined with the mask using the AND operation. Every logical 1 bit in the mask will select a corresponding bit in the word being tested; every logical zero bit in the mask forces a zero in the result independent of the word being tested.

Medium: A plot or a display is usually performed on a two dimensional object which can be viewed by a human being. In the context of this article, the medium is the piece of paper or display tube on which you see the resulting characters.

Page: In many microcomputers it is convenient to divide memory into blocks of multiple bytes, called pages. In the context of this article, the Intel 8080 and 8008 definition is intended: a block of 256 bytes whose high order address byte is identical.

Plotting frame: The range of possible positions for the plotting or display mechanism. In most equipment, this is a grid of points specified by two integer coordinates for horizontal and vertical position.

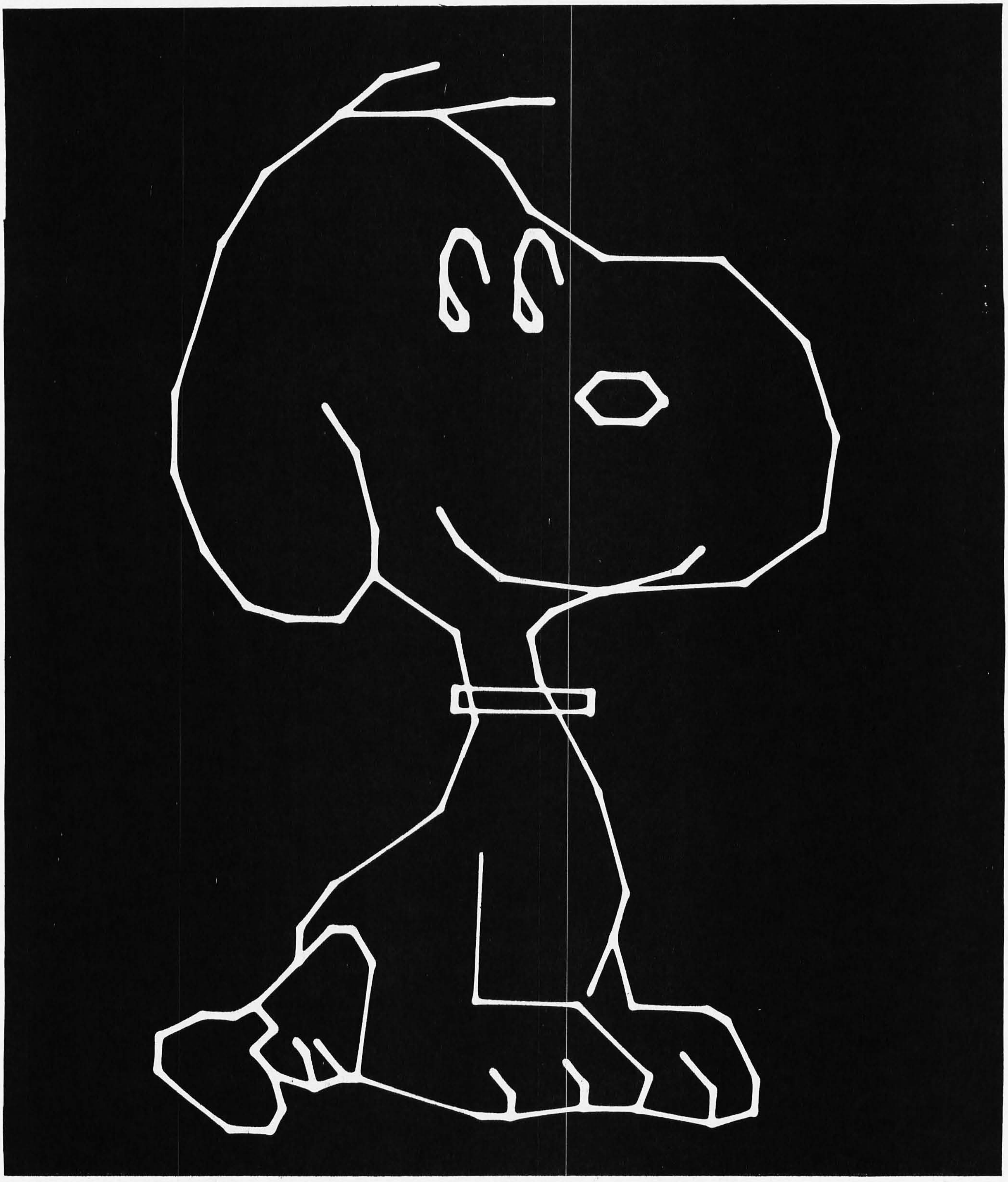
Relative origin: A local origin which is used for convenience of programming. The relative origin is specified by a coordinate pair with respect to an absolute origin of the mechanism used; movements involved in plotting a character are specified with respect to the relative origin to simplify placement of character patterns.

Resolution: A degree of detail involved in the plot. Ultimately this is limited by the resolution of hardware, which is specified as the number of points per linear inch (or centimeter) of display in each coordinate direction.

Subchain: In a chained plotting table, a subchain is like a subroutine of a computer program. It is a fragment of a plot which is often referenced, so use of the subchain economizes the memory requirements of the data tables.

Vector move: In the context of this article, a vector is a line segment which connects two points in the plotting frame. A vector move is the act of moving the plotting mechanism (pen or electron beam) from one of the points to the second point. In a chained approach, as used in this article, the starting point is implied by the last position of the mechanism and the ending point of the move is specified by the coordinates of the position.

Good Grief!



Dave Brockman, 11648 Military Rd S, Seattle WA 98168, has a fairly sophisticated home brew system which includes a vector graphics output capability implemented with a surplus CRT display, heavily modified and interfaced to his PDP-8/S. One result is this vector graphics representation of a famous personality which was taken using a 35 mm camera with multiple exposures to compensate for the slowness of the film used. A program was written to paint a oneshot picture on the screen in response to an input keystroke. The program and camera were set up, the computer room was darkened with blackout curtains, the camera shutter was opened, the oneshot picture-drawing program was cycled several times to burn the image onto the film, then the shutter was closed.

Continued from page 29

should accept the character from the keyboard, then force lower case letters to be upper case letters. In order to avoid interference with the special meanings of the values octal 200 to octal 377, the high order bit should be forced into a zero state by an AND operation with octal 177. After this processing, the input value should be left in the accumulator prior to return.

The character display routine, CHRPR, accepts a seven bit ASCII code which is passed from the calling routine in the accumulator. The characteristics of CHRPR will vary from device to device and it is the responsibility of the person using this program to provide an appropriate CHRPR. The original Educator-8080 program was designed to be used with a Digital Group TV-Monitor output device which displays 16 lines of 32 characters. This device automatically begins a new line whenever the previous line has been filled, so the user of a Teletype or a video display with line length greater than 32 should generate the equivalent of a carriage return and line feed at the end of each 32 character line. This can be done by setting a software counter to 32 at the start of a line, and decrementing every time a displayable character is reached. When the counter reaches zero, the line feed and carriage return codes are sent and the counter is reset to the starting value.

There is one special character value, octal 177, which must be detected by CHRPR and interpreted if necessary. This is the character used for clearing the screen of a video display, or doing a form feed operation on a printing device. For a Teletype device, the form feed operation can be simulated by spacing out the listing. Simply emit a carriage return followed by several line feeds. For a video display communications terminal, the ASCII form feed character, octal 014, is often interpreted as a command to clear the screen. For the Digital Group TV-Monitor display the screen is cleared then the screen address is set for the upper left hand corner.

The Educator-8080 is very useful to the novice. On the other hand, its implementation requires a skill level beyond that of the beginner, who might fare well to obtain the help of an advanced hobbyist in implementing the program.

For The Digital Group 8080 system, a version of this program is available on a cassette including documentation — contact The Digital Group Software Systems Inc, POB 1086, Arvada CO 80001.■

Table 5: Work Areas. The Educator-8080 program employs several work areas in programmable memory. These work areas are listed in this table, along with initial values where applicable, and commentary. Not shown in this table is one very important work area which is part of listing 1: the location XQTOP (address <2>/046) identifies a two byte area which must be programmable to allow an 8080 operation code to be "dummied in" and executed based upon the user's input command.

Address	Name	Length	Octal Initial Value	Commentary
<3>/346	PSWA	2	000 000	Initial PSW and A values.
<3>/350	BANDC	2	000 000	Reference to B and C symbolically.
<3>/350	CREG	1	–	Equivalent to BANDC, C value.
<3>/351	BREG	1	–	Equivalent to BANDC+1, B value.
<3>/352	CMDAR	22	–	Command work area, uninitialized.

What's New?

New Products Policy

Insertions in this section of BYTE are chosen at our discretion from unsolicited "press release" materials sent by vendors. New companies inexperienced in the game of publicity seeking are advised that a press release on a hardware product should contain a descriptive text plus glossy black and white photograph (or photographs) of the item. Software products should include ample descriptive text to convey an idea of what is being marketed; if there are any characteristic graphic materials associated with the software product, pictures or camera ready copy should also be supplied.

Appearance of descriptive information on a product in these columns does not constitute an endorsement of that product.■

Itty Bitty Computers Tiny BASIC for the 6800

Tom Pittman is a professional software person who has developed a debugged and running version of Tiny BASIC for the 6800. Paper tape and documentation are available for $5. The object of Mr Pittman's enterprise, Itty Bitty Computers, is to perform "an experiment in favor of the hobbyist." In a letter to BYTE, Mr Pittman points out that his regular customers are commercial custom software purchasers, who pay the usual high costs of custom software; at $5 for a high level language, users will get quite a bargain since you will receive a fairly thick paper tape and a 24 page computer printed instruction manual. To order Tiny BASIC for the 6800, send $5 to Tom Pittman, PO Box 23189, San Jose CA 95153.■

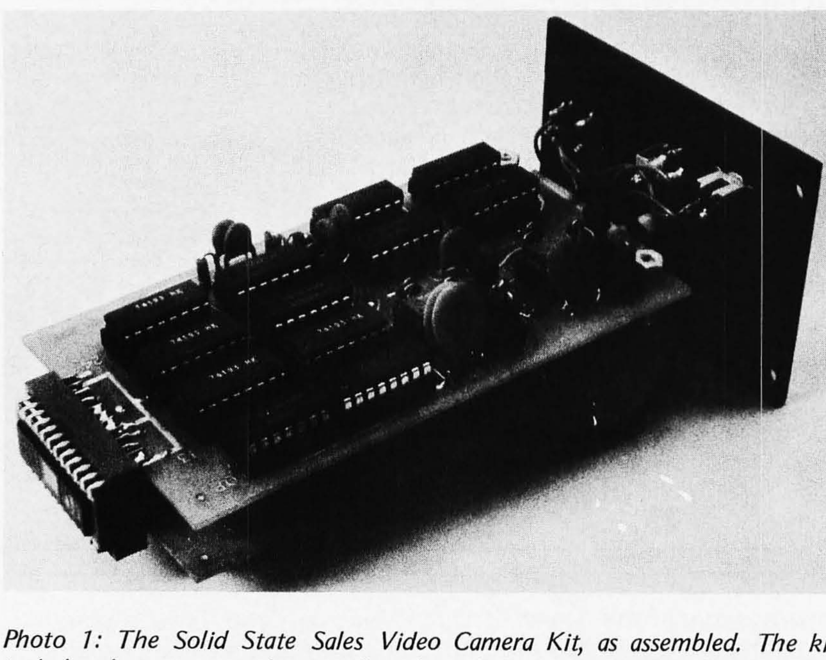


Photo 1: The Solid State Sales Video Camera Kit, as assembled. The kit includes the two printed circuit boards and all electronic parts, but does not include the packaging into a neat box with lenses, etc. The charge coupled device sensor is shown mounted between the two boards in this photograph.



Photo 2: The video signal out of the Solid State Sales Video Camera Kit is shown here on an oscilloscope. This sort of camera could be used as an inexpensive starting point for software experimentation in pattern recognition using personal systems.

Will You Look at That?

Solid State Sales, PO Box 74B, Somerville MA, has introduced a video camera kit which uses a 100 x 100 resolution charge coupled device to produce an EIA video signal which can be used to drive a standard video monitor. The product comes in the form of a kit of parts including the imaging array, printed circuitry and electronic parts. Since the solid state imaging array does not require any high voltages, the camera is an ideal product for experimentation with video and as a source of video signals to be converted to digital form in the context of pattern recognition experiments. The spectral range of the charge coupled device sensor array starts in the infrared region and includes the visible wavelengths of light, so in security applications it could be used with an invisible infrared light source. The price of the kit is $225 including the circuit boards and sensor array as shown in photo 1. For pattern recognition applications a simple converter circuit is needed to convert the video into a binary (black and white) TTL signal with an adjustable threshold. Also, depending upon the application, the user must supply a case and the optics needed to form an image plane at the surface of the sensor array. An example of a digitized signal displayed with the help of an oscilloscope and some external timing circuitry is illustrated in photo 2.■

FREE!

New CAMBION XQ catalog. Scores of famous CAMBION electronic components for hobbyist or experimenter in small packs or individually at modest cost. Parts available at retail electronic shops or by mail from factory. Coils - chokes - IC sockets - tools - patch cords, and lots more. Circle Readers Service number now.

CAMBION

Cambridge Thermionic Corp.
445 Concord Avenue,
Cambridge, Massachusetts 02138
Phone: 617-491-5400

WANTED

Microprocessors Kit or Assembled
ALTAIR • SPHERE • MOTOROLA
INTEL & OTHERS

Memories, I/O boards, chassis, power supplies, modems, peripherals, specials, prom programmers, ADCS, DACS, GPIOS, RS232 faces, and any associated software.

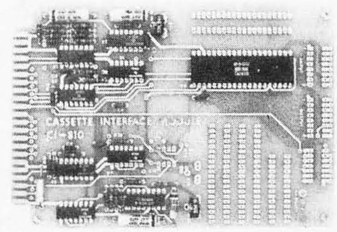
FORWARD INFORMATION AND/OR INQUIRIES TO:
AMERICAN USED MICROPROCESSORS, EQUIPMENT & SUPPLY CORP.
BOX 515,
PRAIRIEVIEW, ILLINOIS 60069
TELEPHONE (312) 634-0076

For complete monthly catalog of items available and in stock Enclose $10. for six issues. Many new unused items. Amazing values for everyone. A-70

TIRED OF LOADING YOUR COMPUTER BY HAND?

TURN AN AUDIO CASSETTE RECORDER INTO A FAST RELIABLE PROGRAM LOADER AND MASS DATA STORAGE UNIT WITH THE *PerCom* CASSETTE INTERFACE FOR $59.95

- Fully assembled and tested
- BYTE Standard (see Feb '76 BYTE)
- Load a 1K program in 40 seconds
- May be upgraded to 218 bytes/sec
- Store 100K bytes on C 60 cassette
- Standard 8-bit parallel interface
- Easily connected to any computer
- Little or no software required
- Provision for custom interfaces
- Signal level sensor
- Sidetone circuit
- Operate 2 tape units simultaneously
- Standard sized circuit card (4½" x 6½")
- Described in BYTE (March-April '76)
- Comprehensive Instruction Manual



TO ORDER: Send check or money order for $59.95. Texas residents add 5%. MasterCharge, BankAmericard welcome, include all numbers and sign your order. Allow 2-3 weeks for delivery, personal checks take longer. Full refund if not satisfied.

PerCom – 'peripherals for personal computing'

Custom Design Services • 4021 Windsor
Garland, Texas 75042 • (214) 276-1968

6 TV GAMES ON A CHIP

Single MOS/LSI Chip
$39.95 each - supply limited
Kit $69.95 (includes chip)
or send $1.00 for kit info
and data sheet
21L02 (500 ns) 1.95
2102 (500 ns) 1.75
8080A 24.95
6800 27.95
AY51013A 5.00
Add $1.00 to cover handling on orders under $25.00.
$10.00 Min. Order.
Calif. residents add 6% tax.

Advanced Microcomputer Products
P. O. Box 17329
Irvine, CA 92713

ON_LINE

A CLASSIFIED ADVERTISING NEWSLETTER DEVOTED TO THE COMPUTER HOBBYIST

ARE YOU LOOKING FOR ?

- LOW COST USED COMPUTER EQUIPMENT
- " " NEW " " " .
- ADVANCE INFORMATION ON NEW PRODUCTS
- SOFTWARE - ASSEMBLERS, SCIENTIFIC,ETC
- A WAY TO ASK FOR INFORMATION OR AID
- COMPUTER CLUB MEETING SCHEDULES

IT'S ALL IN ON_LINE !

HOW OLD IS THIS AD ??

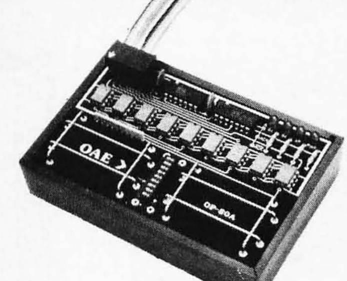
April 20th was when BYTE needed this ad (Other magazines have similar schedules)

WHAT'S THE DATE NOW ?

If ON LINE's deadline were April 20th, that issue would be in the FIRST CLASS mail on its way to you April 24th! And, we publish every three weeks!

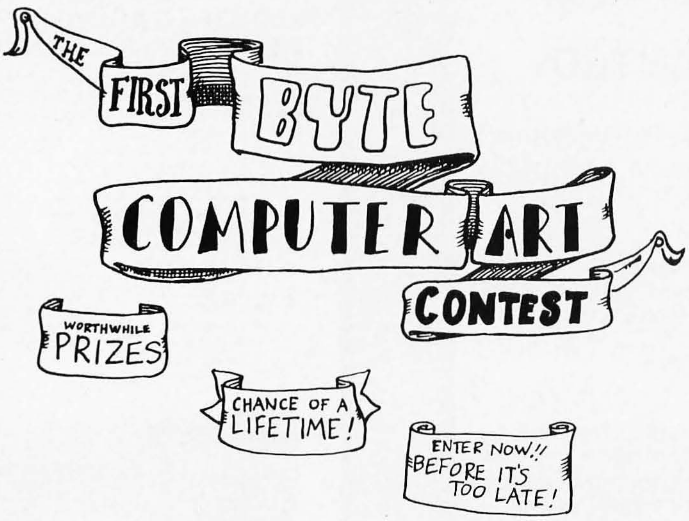
18 ISSUES (1 Yr.)-$3.75 36 ISS.-$7.00
Sample on request, or added to subscrip.

24695 SANTA CRUZ HWY.
LOS GATOS, CA 95030



Pictured above is the new OP-80A High Speed Paper Tape Reader from OAE. This unit has no moving parts, will read punched tape as fast as you can pull it through (0-5,000 c.p.s.), and costs **only $74.50 KIT, $95.00 ASSEMBLED & TESTED.** It includes a precision optical sensor array, high speed data buffers, and all required handshake logic to interface with any uP parallel I/O port.

To order, send check or money order (include $2.50 shipping/handling) to Oliver Audio Engineering, 7330 Laurel Canyon Blvd., No. Hollywood, CA 91605, or call our 24 hr. M/C-B/A order line: (213) 874-6463.



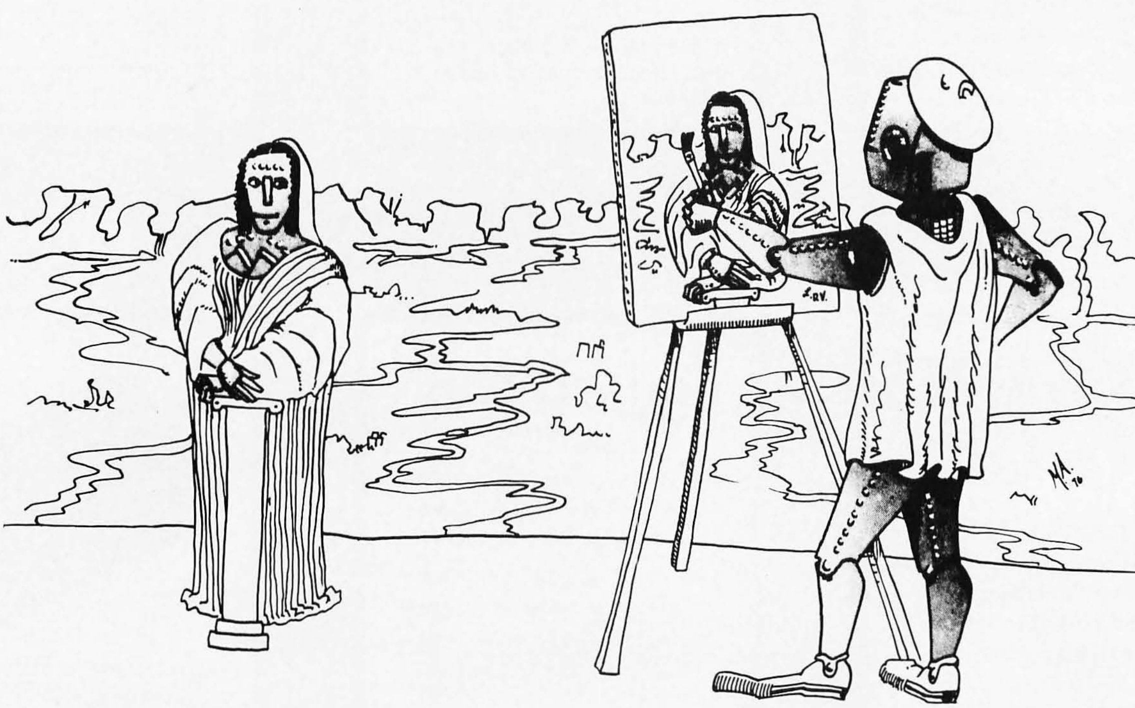
PRIZES

Grand prize winner will receive, in addition to the fame of having his entry appear on the cover of the December 1976 issue of BYTE,

- $100
- A lifetime subscription to BYTE
- A bound volume of BYTE Volume 1.

Two runners up will receive $100 and a five year subscription to BYTE.

Five honorable mention winners will receive a tee shirt wardrobe (one from *Creative Computing,* one from *Computer Lib,* and one from BYTE) and a one year subscription to BYTE.



COMPUTER ART CONTEST

BYTE announces a contest for artwork suitable for use on the cover of BYTE Magazine. Entries can either be artwork created by a computer or human produced artwork on a computer theme.

All entries become the property of BYTE Publications but will be returned if accompanied by a self addressed stamped envelope. Though only one grand prize winner will be chosen to appear on the cover of the December 1976 issue of BYTE, any entries which the BYTE staff like well enough to use as covers will be purchased at our usual rates.

All entries must be received by August 31 1976.

Mail to:

Ms Janice Black
Contest Editor
BYTE Publications Incorporated
70 Main Street
Peterborough NH 03458

BYTE'S BITS

Some Electronics Industry News

For those who are not familiar with the electronic engineering marketplaces, *Electronic News* is one of the best places to find out about late breaking developments in the hardware end of computing. We refer readers to the May 10th edition of that publication, which had the following interesting tidbits:

Page 65: Motorola Semiconductor will shortly (early 1977) be sampling the new M6900 processor. This is described as a 16 bit processor aimed at the computer industry. In short it should be like the central processor of a minicomputer.

Page 66: Mostek and Zilog have made a second sourcing agreement for the new Zilog Z-80 super 8080 processor. [BYTE has scheduled a Microprocessor Update article on the processor for August 1976, to be followed by Dr Robert Suding's detailed central processor hardware design in the September 1976 issue.]

Page 68: Motorola will have a 16 K dynamic programmable random access memory in a 16 pin package, to be sampled later this year. The pinout is compatible with similar devices from TI and Mostek. [It is said that the 16 pin dynamics are pin compatible with the present 16 pin 4 K devices.]

For the details, reference *Electronic News.* The address of the publication is Fairchild Publications, 7 E 12th St, New York NY. Subscriptions to this weekly paper are $12 per annum.■

SOFTWARE AVAILABILITY NOTE:
Perspective Plot Package
(in ANSI FORTRAN Language)

The subroutine package contains 22 calls needed to generate a perspective plot of a three dimensional object or scene described by coordinate points and line segments between such points. The package uses two external calls to the Cal Comp's basic drum plotter system. These calls can be converted very easily to any other system.

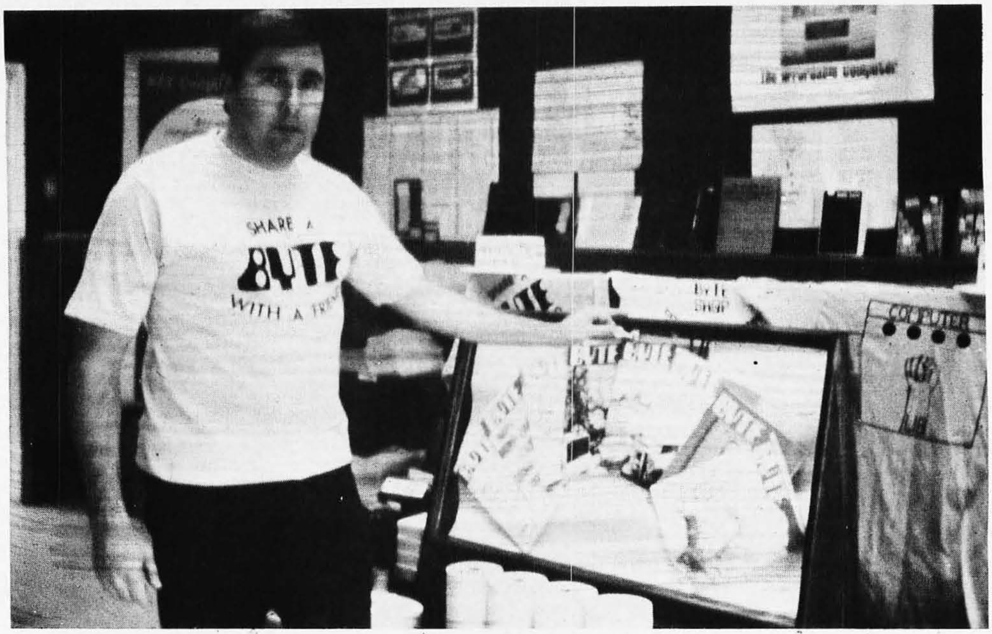
This package is being made available, free of restrictions and cost (except copying and mailing charge), to the individual hobbyist, clubs, schools, colleges, and universities for their own computer systems.

Any commercial use of the package in any part, form, or media for sale as Software or given as an inducement to selling other Software or Hardware is strictly forbidden without the expressed permission of the author.

Copying and mailing cost for each copy is $5, which includes:

1. Users Manual – 37 pages.
2. Hard copy listing of Subroutines – 23 pages.

Richard J Lerseth
8245 Mediterranean Way
Sacramento CA 95826
(916) 381-0335 (after 5:00 PM)



BYTE UNDER GLASS

Paul Terrell, founder of the BYTE Shop, the world's first computer store franchise, puts lock and key to "the family jewels." Each BYTE Shop Computer Store has on display, as a distinguishing mark of its identity, BYTE magazines under glass. The BYTEs were placed under glass as a result of an unfortunate rip off of issue number one after a BYTE subscription ad entitled "A Lesson in Economics" mentioned that number ones were selling for $15.

Though current issues of BYTE are not yet as rare as the first few, they soon will be, as we do not print a large overrun. Don't miss any. Subscribe today. Use the pull out card in this issue or the coupon below.

*** * * * * * SPECIAL * * * * * ***

Subscribe to BYTE and Creative Computing and save $2.

Since we feel that you really need both magazines to be fully informed in the small systems field, the publishers are offering a special package. Receive one year of both BYTE and Creative Computing, normally $20, for only $18. A ten percent saving.

Please enter my subscription to:

☐ BYTE 1 year $12, 2 years $22, 3 years $30 (circle one)

☐ BYTE and Creative Computing 1 year $18, 3 years $46

☐ Check enclosed

☐ Bill me

☐ Charge to MC# ________________ BAC# ________________

Exp. date ________________

Name ________________

Address ________________

City ________________ State ________________ Zip ________________

BYTE Publications Inc **70 Main St** **Peterborough NH 03458**

Please allow six weeks for processing.

Software Bug of the Month 2

A programmer was trying to learn how to write DO loops without using DO statements, and was getting exceedingly frustrated. Every other time he wrote a loop, there would be either one case too many, or one case too few.

As an example, he was trying to print out a table of sines, cosines, and tangents, just as one would find in trigonometry tables. He was doing it by degrees, from one degree up to 90, and somehow the program always ended up printing an extra case, for 91 degrees.

Our programmer knew that the sine, cosine, and tangent functions work with radians, not with degrees. He therefore started with the number that is equivalent to one degree in radians, and worked upward from there. Thus his FORTRAN program was as follows:

```
      PI = 3.1415926
      DEGREE = PI/180.0
      X = DEGREE
    1 Y = X/DEGREE
      WRITE (6,91) Y, SIN(X), COS(X), TAN(X)
   91 FORMAT(F6.2,3F15.8)
      X = X + DEGREE
      IF (X.LT.PI/2+DEGREE) GO TO 1
      STOP
      END
```

The statement Y = X/DEGREE converts X from radians back to degrees, in order to make the table come out with a value in degrees.

Can you find the bug?

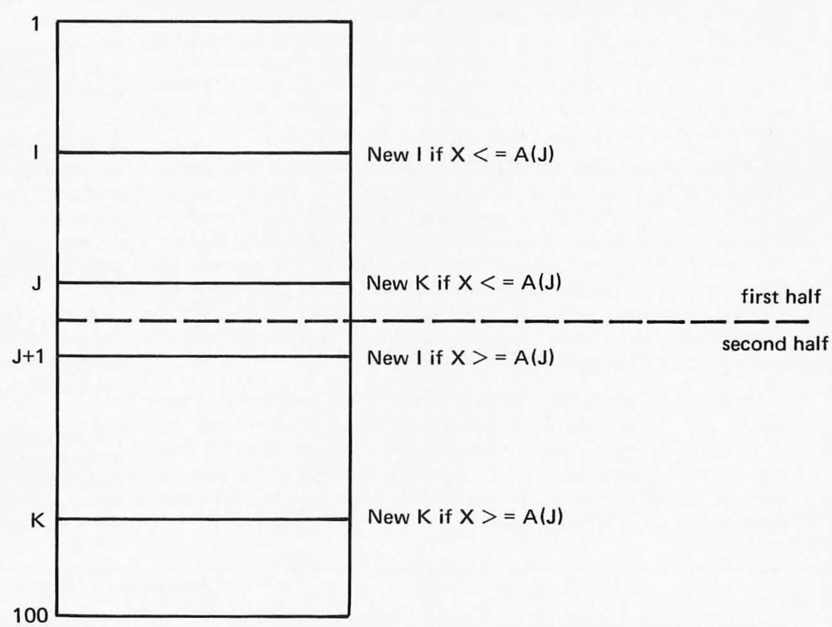
Solution in next month's BYTE

SOLUTION TO BUG OF THE MONTH 1

The problem is *not* in the statement J = (I + K)/2. It is true that I + K might be an odd number, and thus (I + K)/2 might not be exactly half of I + K; but this, as it turns out, doesn't matter.

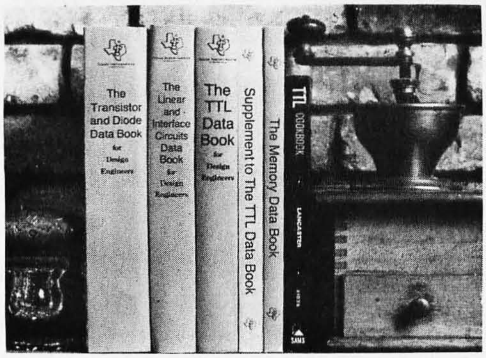
The problem is in the figure, which was drawn in a misleading manner. At the right is the figure as it should have been drawn.

The element with index J is the *last* element of the *first* half of the table. The *first* element of the *second* half of the table is therefore the element with index J + 1, not J. Therefore the new value of I, if X > A(J), is J + 1, and not J. We should change statement number 2 to read I = J + 1.



What happens if the bug is not fixed can be seen, for example, if X = A(2). Eventually we will get down to I = 1 and K = 2. Now J is set equal to 1, which is all right in itself; we test X against A(1), and find that it is smaller than A(1). But then I is set equal to J, which is 1, exactly the same as it was before. This causes the endless loop we mentioned. You can check that it does not occur for *any* possible value of I for which X = A(I) (which ones does it occur for?).■

W Douglas Maurer
University Library Room 634
George Washington University
Washington DC 20052



You'll Want to Nybble at these Byte Books

Where does the editor of a computer magazine turn to when he must verify some author's hardware design? Information on a 75450 interface gate, or a 74147 priority encoder circuit does not spring forth by magic. Checking the information supplied by authors is part of BYTE's quality control program.

When you build a project, you need this same sort of information. All you find in the advertisements for parts are mysterious numbers identifying the little beasties . . . hardly the sort of information which can be used to design a custom logic circuit. You can find out about many of the numbers by using the information found in these books. No laboratory bench is complete without an accompanying library shelf filled with references – and this set of Texas Instruments engineering manuals plus Don Lancaster's *TTL Cookbook* will provide an excellent starting point or addition to your personal library.

- **The TTL Cookbook** by Don Lancaster, published by Howard W. Sams, Indianapolis, Indiana. Start your quest for data here with Don's tutorial explanations of what makes a TTL logic design tick. 335 pages, $8.95.

- **The TTL Data Book for Design Engineers,** by Texas Instruments Incorporated. How does an engineer find out about the TTL circuits? He reads the manufacturer's literature. This 640 page beauty covers the detailed specs of most of the 7400 series TTL logic devices. No experimenter working with TTL has a complete library without **The TTL Data Book for Design Engineers.** Order yours today, only $3.95.

- **The Supplement to The TTL Data Book for Design Engineers,** by Texas Instruments Incorporated. What happens when you can't find a 7400 series device listed in **The Data Book for Design Engineers?** Before you start screaming and tearing your hair out in frustration, turn to the Supplement. The Supplement has 400 pages of additional information including a comprehensive index to both TTL Data Book volumes. To be complete (and keep your hair in place and vocal cords intact) you'd best order the supplement at $1.95 to accompany the main volume.

- **The Linear and Interface Circuits Data Book for Design Engineers,** by Texas Instruments Incorporated. When you run across one of those weird numbers like 75365 the immediate frustration problem occurs again. What kind of gate could that be? We won't tell in this ad, but you can find out by reading the specifications in **The Linear and Interface Circuits Data Book for Design Engineers.** You can interface your brain to the 72xxx (linear) and 75xxx (interface) series of functions by ordering your copy of this 688 page manual at only $3.95.

- **The Semiconductor Memory Data Book for Design Engineers,** by Texas Instruments, Incorporated. Don't forget the importance of memories to your systems. Refer to this 272 page manual to find out about the T.I. versions of many of the popular random access memories and read only memories. Order your personal copy today, only $2.95.

- **The Transistor and Diode Data Book for Design Engineers,** by Texas Instruments, Incorporated. You'd expect a big fat data book and a wide line of diodes and transistors from a company which has been around from the start of semiconductors. Well, its available in the form of this 1248 page manual from T.I. which describes the characteristics of over 800 types of transistors and over 500 types of silicon diodes. This book covers the T.I. line of low power semiconductors (1 Watt or less). You won't find every type of transistor or diode in existence here, but you'll find most of the numbers used in switching and amplifying circuits. Order your copy today, only $4.95.

- **The Power Semiconductor Handbook for Design Engineers** by Texas Instruments, Incorporated. To complement the low power transistor handbook, T.I. supplies this 800 page tome on high power transistors and related switching devices. Here is where you find data on the brute force monsters which are used to control many Watts electronically. Fill out your library with this book, available for only $3.95.

- **Understanding Solid State Electronics** by Texas Instruments, Incorporated. This is an excellent tutorial introduction to the subject of transistor and diode circuitry. The book was created for the reader who wants or needs to understand electronics, but can't devote years to the study. This 242 page softbound book is a must addition to the beginner's library at only $2.95.

- **The Optoelectronics Data Book for Design Engineers** by Texas Instruments, Incorporated. This 366 page book is a compendium of information on T.I. phototransistors, LEDs and related devices. Order yours at $2.95.

Buyers of these books should be cautioned: heavy reading will be required. These books are so filled with information that they weigh in at a total of about 190 ounces (5387 grams). On the basis of sheer mass, these books have got to be the bargain of the century. Make sure that you use a structurally sound book shelf and above all avoid dropping one of these books on your foot.

____ TTL Cookbook @ $8.95
____ TTL Data Book @ $3.95
____ Supplement to TTL Data Book @ $1.95
____ Linear and Interface Circuits @ $3.95
____ Semiconductor Memory Data @ $2.95
____ Transistor and Diode Data Book @ $4.95
____ Understanding Solid State Electronics @ $2.95
____ Optoelectronics Data Book @ $2.95
____ Power Semiconductor Handbook @ $3.95

Please add 75 cents for postage and handling. Please allow six weeks for delivery.

Send to: Name ________

Address ________

City ____ State ____ Zip ____

☐ Check enclosed

☐ Bill MC # ____ Exp. Date ____

☐ Bill BA # ____ Exp. Date ____

Signature ________

BYTE PETERBOROUGH, NH 03458

What Happened at Trenton May 2

Photos by Marj Kirk

On May 2 1976, the Amateur Computer Group of New Jersey held the first Trenton Computer Festival, a combination of manufacturers displays, flea market activities, and technical presentations. The site of the affair was Trenton State College, in Ewing Township NJ. The photos give several vignettes of the activities.

There were a total of 45 exhibitors present, and 25 speakers in the technical program. The attendance was counted by that effective method of selling tickets which registered guests for the door prize drawing. Of the 1500 people who purchased tickets at the door, 180 received door prizes at the end of the day. People from 16 different states attended.

Despite a torrential downpour (which set local records) the day before, the skies cleared for a beautiful sunny festival day. This made the outdoor flea market activities a big success. Items on sale at the flea market moved quite briskly, and ranged in price from 25 cents or less to the $10,000 asking price one individual had on his original Samuel FB Morse telegraph key.

Exhibitors booths included firms ranging from computer stores to such firms as Motorola Semiconductor Products, Digital Equipment Corporation, and RCA Solid State Division.■

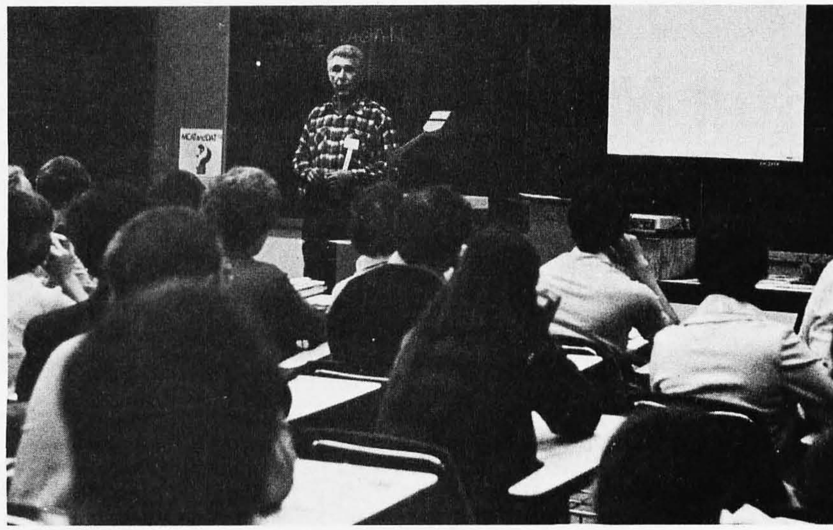
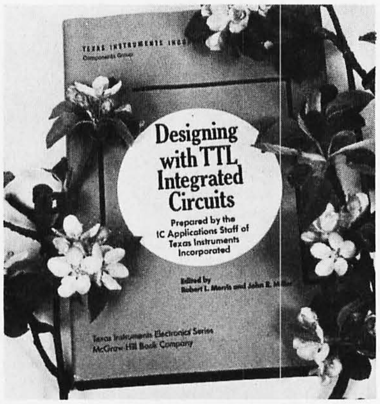


Photo 1: Claude Kagan, a researcher at Western Electric, gives a talk on "Computers in the Home, Present and Future."

More Trenton photos on page 85.

Still More BYTE's Books





•DESIGNING WITH TTL INTEGRATED CIRCUITS by the Components Group, Texas Instruments Inc. Edited by Robert L Morris and John R Miller.

People often ask questions like "Where do I get basic information on hardware design?" One answer is in **"Designing With TTL Integrated Circuits."**

This book, published by McGraw Hill in 1971, is a fundamental starting point for any person designing peripherals and custom logic employing TTL integrated circuits. While its publication date precludes any reference to the later additions to the TTL 7400 series of components found in the **Data Books,** it is nevertheless the source of a wealth of ideas on TTL integrated circuits and design of logic with this family of circuits.

What is fanout? You may have heard this term mentioned at computer club meetings or in advertisements for circuitry, or in articles in BYTE. You can find out background information on the calculation of fanouts by reading the chapter on **Circuit Analysis and Characteristics of Series 54/74.**

Worried about noise, shielding, grounding, decoupling, cross talk and transmission line effect? (Or, more properly, did you know you should worry about these effects in certain circumstances?) Find out about general precautions and background information by reading the chapter on **Noise Considerations.**

The chapter on **Combinatorial Logic Design** gives 53 pages of background information on Boolean algebra and practical representations of logic in the form of SSI gates. The chapter includes a description of Karnaugh mapping techniques and the minimization of logic. From combinatorial design, the book progresses into **Flip Flops,** including background information on the workings of these devices, and fairly detailed descriptions of the uses and applications of these devices including synchronization of asynchronous signals, shift registers, flip flop one shots, etc. Then the book returns to static combinatorial logic with its description of the **Decoders** available in the 7400 line as it stood in 1970-1971.

A chapter on **Arithmetic Elements** gives fundamental descriptions of binary arithmetic, diagrams of the basic gate configurations for combinatorial logic adders, and a section on number representations for use in computers. Much of the material in this section is dated, due to the fact that the later 74181 series of multiple function arithmetic units had not yet appeared when the book was written. But for a background on arithmetic operations implemented with the simpler 7483 circuits, this chapter is ideal. A chapter on **Counters** and a chapter on **Shift Registers** complete the detail logic sections. The book is closed out by a chapter on miscellaneous **Other Applications** including a simple binary multiplier, a 12 hour digital clock and a modulo-360 adder.

The most important use of this book is its value as an introduction to TTL logic. By reading and studying it, you will begin to understand the ways in which SSI and MSI TTL gates can be utilized in your own experimental logic designs. After studying this text, you should be able to make much more sense out of the technical information summaries typically published as specifications sheets and data catalogs.

Order your copy today from BYTE's Books, $24 postpaid.

•MICROCOMPUTER DESIGN by Donald P Martin, Martin Research. Edited and Published by Kerry S Berland, Martin Research.

Purchase your copy of the definitive source for circuitry and hardware design information on the 8008 and 8080 computers today.

Even Intel, the originator of the microprocessor revolution, is hard put to compete with the wealth of information found in Martin Research's new second edition of **Microcomputer Design.** This is the book which was originally published as an expensive (but quite practical) engineering report in loose leaf form, at about the time the microprocessor technology was first catching on in the form of the 8008. This 388 page second edition of the manual is loaded with detailed information on how to build and use computers based on the 8008 and 8080.

But even if you do not intend to use the 8008 or 8080, the practical pointers on digital logic design, peripherals and applications of hardware techniques will more than justify the new low price of $25 for this handbook. **Microcomputer Design** is a must for 8008 owners and 8080 owners who want to truly understand how their processors process.

Microcomputer Design is complete with numerous illustrations, tables and diagrams, plus reprints of the specifications sheets for the Intel processors. There are numerous practical examples of circuitry and many complete computer designs ranging from "minimal microcomputers" to a full blown 8080 processor.

Order your copy today, $25 postpaid from BYTE's Books.

BYTE PETERBOROUGH, NH 03458

___ Designing With TTL Integrated Circuits $24

___ Microcomputer Design $25

Send to: Name ______________________

Address ______________________

City ______________ State ________ Zip ________

☐ Check enclosed

☐ Bill MC No. ______________ Exp. Date ________

☐ Bill BA No. ______________ Exp. Date ________

Signature ______________________

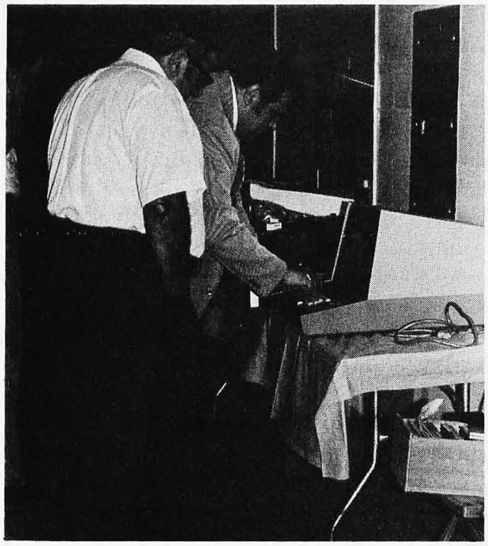


Photo 2: Herb Nelinger of Miami FL gets a demonstration of one of the Delta Data CRT terminals which were sold at the festival. To his complete surprise, Kevin Moran of Delta Data sold out all his terminals practically before the doors had opened as the ACGNJ people running the show snapped them up.

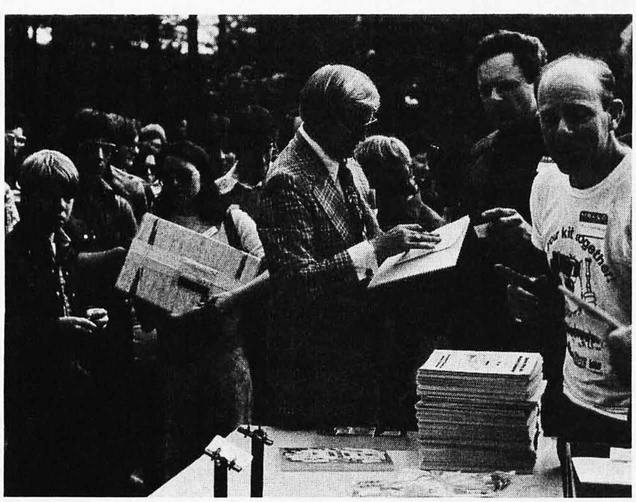


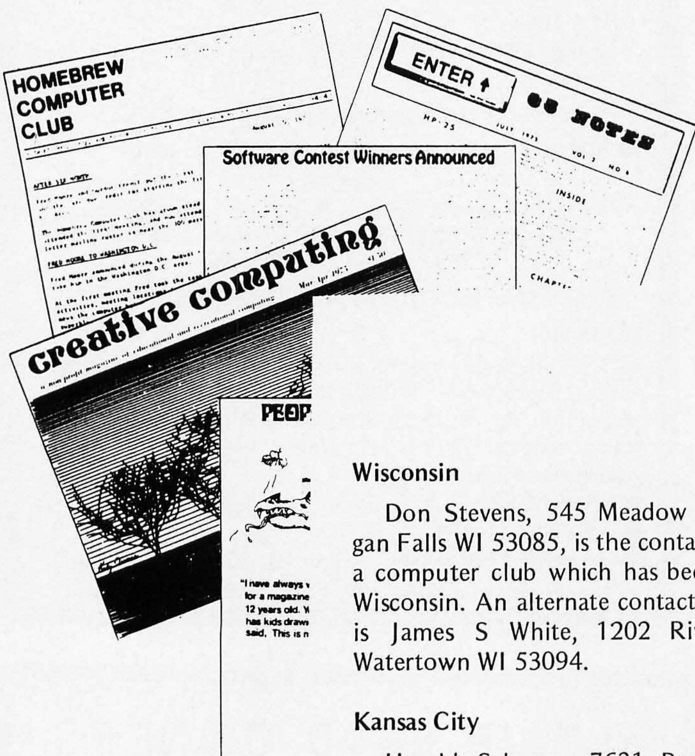
Photo 3: For some (namely the lucky winners) the drawing at the end of the day's activities was the high point. The young lady holding the cardboard box has won an MOS Technology KIM-1 processor board which was one of the 180 door prizes given out at the festival.

Clubs and Newsletters

SCCS Personal Computing '76 trip.

The Southern California Computer Society has made arrangements with Travel Coordinators, 8317 W Third St, Los Angeles CA 90048, for a group tour travel arrangement for individuals attending Personal Computing '76, August 28 and 29. The show looks like a big event, and will be held in Atlantic City NJ with representatives of most manufacturers in the field purchasing booth space and contributing door prizes.

The tour basis group rate air fare will be available through Travel Coordinators from any major city in the US. For information write Personal Computing '76 Trip, Travel Coordinators, 8317 W Third St, Los Angeles CA 90048, or call (213) 655-0650. To make a reservation by phone call (213) 655-0650 collect.



Wisconsin

Don Stevens, 545 Meadow Ln, Sheboygan Falls WI 53085, is the contact person for a computer club which has been formed in Wisconsin. An alternate contact for this club is James S White, 1202 Riverview Ln, Watertown WI 53094.

Kansas City

Harold Schwarz, 7631 Broadmoor Ln, Overland Park KS 66204, would like anyone in the Kansas City MO and Kansas greater metropolitan area to contact him if interested in forming a computer club. Phone (913) 371-2616 from 9 AM to 4:30 PM or (913) 648-5410 after 6 PM.

Language Processor's Committee?

Here is an idea for a club or interest group defined by a logical classification rather than a geographical one: Robert Heller, Box 51A Star Route, Wendell MA 01379, requests "anyone interested in starting a committee to study the possibility of writing language processors for PL/M or SNOBOL 4 for 8008, 8080 or 6800 systems, please write to me." The idea presumably is to achieve the necessary software by writing processors for the fun of it, and incidentally achieve a time and money sharing effect.

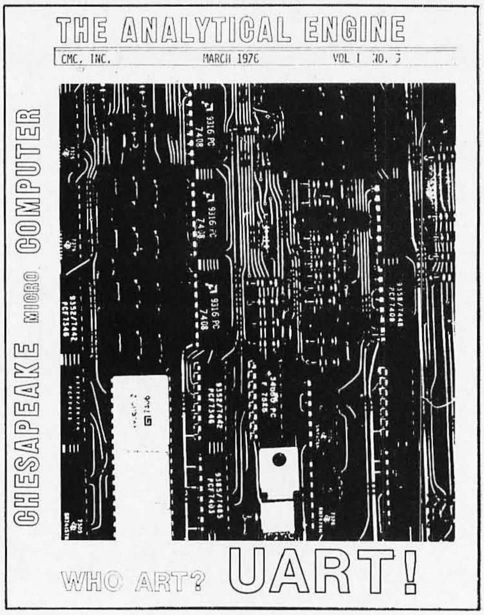
San Diego Computer Society

The latest *Personal Systems* (the SDCS newsletter) to pass our desk was Volume II Issue 1 for March 1976. In this issue was an excellent potpourri of technical and miscellaneous information including: Ralph McElroy's article on how to convert a television into a video monitor as exemplified by a Hitachi portable TV chassis conversion made with the aid of a Sams Photofact File; "Understanding the VDM" by Stan Skoglund, wherein you'll find information on how the Processor Technology VDM module can be programmed and utilized; a reprint of part one of Alan Hastings' "A Software UART" found in *The Analytical Engine* of the Chesapeake Micro Computer Club; a BASIC game called AMAZIN by Frank Maclachlan, "A Microprocessor Survey" by Lance A Leventhal, "Power Supplies and Regulators" by David G Llenaresas, and a Super Nim program by Jim Farschon. The technical content of *Personal Systems* is excellent, and will benefit many readers. This newsletter can be obtained for an SDCS membership at $10 per annum. Contact SDCS, PO Box 9988, San Diego CA 92109, attn: Newsletter.

The Analytical Engine, Continued

Each issue of *The Analytical Engine* put out by the Chesapeake micro Computer Club, 236 St David Ct, X4, Cockeysville MD 21030 (phone (301) 667-9690) is an excellent addition to the literature of personal computing. Here is a club newsletter that is well worth a subscription even if you live in Prudhoe Bay AK. The March issue contained several views about the issue of software piracy and proprietary rights. The major technical contribution of this issue was Carl Hallberg's background article on use of the common garden variety programmable





UART integrated circuits, and a continuation of John A Hastings' article on programming a software UART algorithm. Besides the technical "meat" found in the two asynchronous communications articles, the 16 page *Analytical Engine* includes a flourish of professionalism with halftone photos.

Washington Amateur Computer Society

The nation's capital computer club sent a February edition of their newsletter, Volume 1 Number 1. It is one of the few such papers prepared and reproduced using a line printer. (The dot matrix (5 by 7) upper and lower case listing suggests it is the product of a DECWriter.) The newsletter included the club constitution (brief and to the point), plus a questionnaire. Contact WACS c/o Robert J Jones, 4201 Massachusetts Av, Washington DC 20016.

Are There Any Others in Idaho?

Edwin S Hill, 1900 W Quinn #117, Pocatello ID 83201, would like to meet others who want to form a club to service southeastern Idaho, southwestern Wyoming, northwestern Colorado and northeastern Utah. We know there are a bunch of BYTE readers in that area, so how about a club?

Westchester Amateur Computer Society

The kickoff meeting of the Westchester Amateur Computer Society was April 1 1976 at the Community Room of the Greenburgh Public Library in Elmsford NY. For residents of Westchester County NY interested in further information, call Harold Shair at (914) 967-7853, or write him at 41 Colby Av, Rye NY 10580.

Quest: Philadelphia

Walter White and Everett Holland are interested in making contacts for computer club activities in the Greater Philadelphia Area. Anyone interested, please drop a line to Everett Holland, PO Box 462, Wayne PA 19087, or call evenings at (215) 647-8460.

Re-Quest: Philadelphia

2005AD Inc, Philadelphia PA, wishes to establish a computer club in the metropolitan area. Write 2005AD Inc, 2005 Naudain St, Philadelphia PA 19146. [See also page 70 of February 1976 BYTE for a listing of the SCCS Delaware Valley chapter in Philadelphia/Camden area.]

CACHEing IN?

A quite unofficial contact from Charles Douds reported on some of the activities of the Chicago Area Computer Hobbyist Exchange. In his letter, he mentioned that the March 28 meeting included a speaker from Motorola talking about the 6800 family (what else?) and an IBM salesperson giving "a full blown IBM dog and pony show" about their top of the line (for those with limited budget) IBM 5100 table top APL and BASIC machine. [The 5100 is an interesting beast; word on the technological grapevine has it that the APL and BASIC were implemented by having a microcontroller emulate the System/360 (370) architecture. Then IBM engineers stored the machine language object code of an appropriately patched version of the "big system" APL and BASIC in the monstrously large ROM chips IBM uses.] Also reported in Charles' letter is a suggestion attributed to Ruth Lowe that the name of the CACHE Newsletter be changed to "CACHE Register" on the grounds that nobody ever has enough registers to work with in low level code.

Individuals interested in finding out about CACHE should write CACHE, PO Box 36, Vernon Hills IL 60061.

Unofficial Purdue University Club – Philanthropists Take Notice

"A group of (insane) Computer Bums at Purdue University is looking for a PDP-11 series mini, along with associated equipment. We are interested in *any* and all hardware which might be useful for same. In trade we can offer a limited amount of cash, and/or custom software work. We are very familiar

with: CDC-6000 series, CDC-1700 series, all PDP series, HP-2100 series, Imlac graphics, PDP graphics, 8008, 8080, 6800, and others, all major languages. If you have spare equipment you don't use and need custom software, we welcome your inquiries. Contact: JGM/TGI/GHG at PO Box 2345, West Lafayette IN 47906 or (317) 463-7167. Philanthropists very, very welcome."

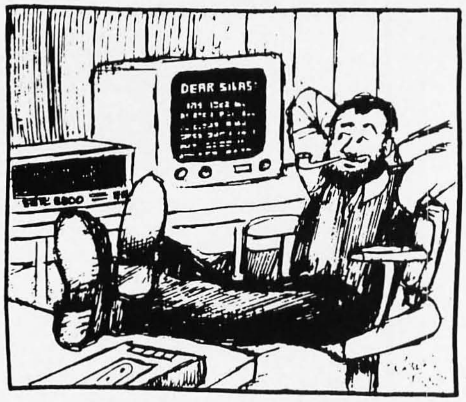
North Alabama Club Activity

Persons in the Huntsville – North Alabama area interested in any aspect of amateur or hobby computing are invited to join the North Alabama Computer Club (NACC). Contact Jack Crenshaw, 1409 Blevins Gap Rd SE, Huntsville AL 35802. Phone (205) 859-7344 or 883-7973.

Mid Michigan Microcomputer Group (M^3G)

Activities in the central portion of Michigan have resulted in the "Mid Michigan Microcomputer Group." For further information, contact William Serviss, president, at 13121 Tucker Dr, DeWitt MI 48820, phone (517) 669-3179, or Daniel Herrick, vice president, at 1214 Frederick St, Box 513, Owosso MI 48867, phone (517) 723-3264.

Letters



THE POLAROID HARD COPY UNIT?

I want to suggest a cheap, portable, multiuse, easy to interface, and inexpensive hard copy unit for BYTErs with video terminals — the camera. I suggest:

1. A multipurpose camera loaded with cheap black and white film for photographic listings.
2. That the camera and film should be compatible with the home production of microfiche (a viewer can be built for $15 — reference on request). A high resolution film will be desirable to make eventual computer scanning of your fiche easier (assuming that you will make fiche of things not in the computer system already — quite likely since costs can be less than 1¢ per page).
3. Personally, I would like to add a Polaroid SX70 for instant color ($1 per picture) copies of Dazzler displays.

Last — since I know nil about photography, would some BYTEr write an article about photographic copy systems for home computers, preferably with all the information a novice needs to get a system up and running, plus cost information for the cameras, film and developing needed?

Martin Buchanan
2040 Lord Fairfax Rd
Vienna VA 22180

AN OPINION ON SOFTWARE MARKETING

Let me first say that I write as a concerned hobbyist. My comments are my own and do not necessarily reflect the opinions of any organization of which I may be a member.

Bill Gates' "An Open Letter to Hobbyists" very clearly explained one of the chief problems of the hobby computer industry, the low Return on Investment (ROI) on the software component of the system sold [for a copy of Bill Gates' views, see page 14 of *Radio Electronics*, May 1976, page 24 of March-April 1976 *PCC* (Box 310, Menlo Park CA 94025), page 3 of February 1976 *Computer Notes* (published by MITS Inc)].

I have no solution to this problem. I only wish to express my personal views on three related aspects of the computer hobbyist market: recovery cost per unit, software delivery methods, and target market expansion.

Recovery Cost per Unit

When was the last time you saw the word "profit" written on a rest room wall? Profit is not a four letter word. It is the incentive that brings vendors into this insane world of home computing. It is no more evil than indirect addressing or fig newtons. Assume that a vendor adopts the following profit philosophy.

Profit will be generated from the hardware sold. Since hardware is of no use without good software, a high initial capital investment will be made in software. This cost will be partially recovered with each unit sold. Recovery cost per unit must be as low as possible to discourage unauthorized duplication of software.

You can see from this that

> (Recovery cost per unit) is equal to (Total software cost) divided by (Number of units sold)

Therefore, to minimize recovery costs, vendors must view the hobby market not in terms of hundreds or thousands, but tens of thousands of units.

One problem with a volume of this size is that present software delivery methods of cassettes and paper tape will not be economical. New delivery methods must be used.

Software Delivery Methods

A high volume software delivery method should be based on proven technology, use agreed upon standards (such as Kansas City), be easily adaptable by all hobbyists, and have a low per unit cost in high volume.

One method that may work, although it's probably not the best, is the use of standard 33 1/3 rpm long playing phonograph records. It meets all of the requirements from above. Interfacing would be done from the turntable to the cassette input. A transformer may be needed for impedance matching. I'm sure other hobbyists will devise better methods.

Now that the need for a high volume of software and a possible delivery method

have been established, the question remains of how to create a market to justify this.

Target Market Expansion

The best approach to increasing the number of computer hobbyists is to publicize our activities in a medium that is accessible to the general public. Once people see what we are doing with a few K of memory, a couple of LEDs, and a lot of sweat, I'm sure it won't be long before they too are "hooked."

I therefore propose that hobbyists, individually and through their local clubs, start writing to their local Public Broadcasting television station suggesting that they join with other stations to produce a special on this "world's greatest toy."

Who knows, we may one day be able to enjoy a weekly half hour program on home computing while our micros are storing the latest game of the week, by frame grabbing, off the same television signal.

To summarize, I believe software costs can be reduced considerably by increasing the size of the market by at least an order of magnitude, and devising new delivery methods to meet that market. It remains for hobbyists to inform the general public of our activities so that all this may happen.

What are you waiting for? Fire up your pencils, pens, 33 ASRs, and matrix printers. Send that letter off now. Address it to the director of programming in care of your local PBS station.

You might even begin it with, "As one programmer to another . . . . "

Robert H Wada
Garden Grove CA

LET'S HAVE A SNOBOL FIGHT (BUT ICEBOLs ARE NO FAIR)

"The Magic of Computer Languages" [April 1976 BYTE, page 24] was a good article, but I wish people would stop writing about "compiling languages" and "interpretive languages." Any language can be either compiled or interpreted. *[Quite true.]* Small machines tend to use interpreters and large machines tend to use compilers. No doubt 4096 people have already pointed out that APL programs are executed from right to left, so I won't mention that. *[?]* SNOBOL has been simplified already; have you never heard of ICEBOL? *[No.]*

"Frankenstein Emulation" [April 1976 BYTE, page 50] was interesting. I suspect that the Main Processor only appears to be faster than the Top Processor because it does

very little computing, operating mainly by table lookup and short library routines. Only the Top Processor seems capable of real time operation. The biggest glitch in the whole system is the Value Judgement Table. The Main Processor was never designed to handle value judgements, but insists on trying. No wonder an external Diagnostic Processor is needed!

Great article on the TMS9900 [April 1976 BYTE, page 64], but a bit too short. How about publishing a similar one on the RCA CDP1800?

J Gordon
Los Angeles CA

A STAR TREK PRODUCT

Several letters have recently appeared in BYTE concerning the history of the Star Trek game. A version of the game is now available for the Altair 8800. It is written in Altair 4 K BASIC and is available from International Data Systems Inc. The purchase price of $10 (checks OK) includes a complete program source listing, operational instructions, tips on how to "patch" the program to add your own features, a one year limited warranty against "bugs," and postage and handling. The limited warranty states that a corrected copy will be provided if any errors are identified. The limited warranty does not provide replacement for time or resources lost as a result of such errors. Orders should be addressed to Star Trek Offer, International Data Systems Inc, PO Box 593001-AMF, Miami FL 33159.

D E Hipps
Vice President
International Data Systems Inc

SPACE WAR ORIGINS UNVEILED

I'd like to answer the comment in AB Bonds' letter (page 8 of April 1976 BYTE) on the "mysterious" origins of Space War. I believe that the first Space War was implemented on a PDP-1 at MIT. This machine was one of the first models of one of the first minicomputers ever made. It was donated by DEC to MIT where it created a new type of student — the "computer hacker." Several members of the Tech Model Railroad Club attached themselves to this machine, developing, besides the original Space War, one of the first timesharing systems, predecessors to DDT and TECO, and I believe the first LISP implementation. This Space War had a sun with gravity, torpedos, and hyperspace. It ran in 4 K words of memory.

Several of the PDP-1 hackers went on to the MIT Artificial Intelligence Laboratory when it started up with a PDP-6. Space War came with them. My introduction to the game at MIT was on the AI lab's PDP-6. That version was similar to the PDP-1 version. One of its new features was an accurate star field in the background (with proper magnitudes) that moved slowly as the game progressed and the observer moved around the sun.

I have since seen a Space War implementation on a PDP-9, a PDP-7 and even on an IBM 7040/7094 DCS. The latter was a large scale second generation batch system at Yale. The only way we could take over the machine to play was with the cooperation of some third shift operators who were also Space War addicts. We would check the control cards of the night's runs to see how much time we could spend playing and still get all the users' jobs done by morning. We would then send out for pizza and start playing.

The best version of Space War I have yet seen was developed on the MIT AI Lab's PDP-6 by a person with the login ID KLH. This is a Star Trek-y game, with one ship looking like the Enterprise and optional use of phasers and deflector shields instead of torpedos and hyperspace. This game also has space mines and invisibility shields.

Sidney Markowitz
Cambridge MA

Then, of course, there are the MIT undergraduate courses in digital electronics which come up with a new variation of Space War as a lab project each year.

MORSE POWER TO YOU

I have just received my April issue of BYTE and regarding Richard Fall's request for information on Morse translation algorithms: I wish him luck. There is a remarkable dearth of information available to the average person. I have been working on my master's thesis and have been researching this area for over a year. I will try to save Richard and others like him the trouble of looking for information that is not really there. Virtually all of the information available today is either military related (that generally means classified!) or is closely guarded by civilian companies that are hoping for government contracts (that generally means classified too, or even worse, proprietary to the company).

There are a few articles on hard wired Morse decoders in amateur radio magazines

from which a simple mathematical model can be derived, but I caution the overzealous experimenter for there is probably no ONE satisfactory algorithm to translate HAND-sent Morse code. Duc to the high degree of variability of mark/space characteristics (for those who don't care for the higher math terms, this means that the lengths of dots, dashes, and the various spacing intervals are constantly changing) ANY fixed algorithm will eventually fall apart and fail to decode properly mainly because we do have some pretty rotten fists on the air today!

Obviously, the more complex the algorithm, the more memory it will require. Some professionally designed and programmed algorithms I have seen on PDP-11-type machines run in the vicinity of 8 K for a BASIC decoder. This figure can go as high as 250 K or more for the more sophisticated models, but in most cases the simpler models will work fine for jisting copy (getting the general trend) and are generally unsatisfactory if your interest is in solid copy.

Machine-sent Morse and also RTTY are completely different problems, since they represent machine-to-machine interface rather than man-to-machine interface. I would advise interested readers to look up: (1) November 1971 *Ham Radio* magazine (2) January 1971 *QST* magazine (3) October-December 1975 *QST* and either or both of the following papers if they are available at your library (try college libraries too):

(1)"On Computer Transcription of Manual Morse," *Journal of the Association for Computing Machinery,* Volume 6 #3, July 1959 (?), pp 429-442, by Charles R Blair.

(2)"Machine Recognition of Hand Sent Morse Code," *IRE Transactions on Information Theory,* March 1959, pp 17-24, by Bernard Gold.

These articles are NOT simple, but should provide some insight to the magnitude of the problem of Morse Code translation.

W A Hickey
Naval Postgraduate School
Monterey CA

TAKE A STAND?

Is the computer hobbyist getting state of the art components? How many years must pass before the hobbyist gets new developments?

EXAMPLE: *Electronics* magazine, January 8 1976, page 76: Matsushita Electrical

BYTE's election year stand on motherhood, apple pie, computer power for the people, and state of the art components for hobbyists.

Industrial Company has developed a floppy disk memory at the price of a digital cassette.

EXAMPLE: Calculator and computer companies are using 4 K RAMs now. Some use 4 K RAMs in designs that permit an exchange for 16 K RAMs when they appear in quantity this summer.

EXAMPLE: *Electronics* magazine, January 22 1976, page 40: Axiom Corp is manufacturing a microprocessor controlled line printer at a cost of $1000 in a single quantity, and $500 in OEM quantities.

When do we get these components in our kit versions?

Where does BYTE magazine stand on this issue of state of the art components hobbyists?

Edward L Tottle
Baltimore MD

To paraphrase Ted Nelson's introduction to his talk at the MITS World Altair Computer Convention: "Motherhood!" "Apple Pie!" "computer power for the people!". Of course the computer hackers are getting state of the art components. State of the art means current technology which both works and is manufacturable and salable. As for timing of specific items based on what you see in electronics engineering magazines, that is up to the entrepeneurial designers who create actual products which can be marketed to the personal computing consumer. So, as with motherhood, apple pie and computer power for the people, BYTE is definitely in favor of advances in the state of the art.

SUPER SERVICE TESTIMONIAL

I don't know how many computer hobbyists are at the hardware experimentation stage. If the computer club here in Columbus OH is any indication most are probably still at the thinking and planning level, but for those who are building I would just like to provide a little "reader feedback" on the service from a couple of your advertisers. All us hardware types are familiar with the "back order blues" that holds up experimentation for months, so the ultra fast turn around service I got from Advanced Data Sciences (they sell keyboxes) and S D Sales (they sell parts) deserves special credit.

Since it was a letter just like this one in an amateur radio magazine that put me onto S D, I just thought a little feedback among computer types might be of a little interest for those who are about ready to buy.

B F Jacoby
88 W Frankfort
Columbus OH 43206

WHERE IS THE INEXPENSIVE GRAPHICS DISPLAY CIRCUIT?

I can't understand it! There are so many computer hobbyists that I know without any IO. And yet we can find no circuit that will take our 8 bit output and put it on our TV sets.

There are many TV typewriters, TV games, TV graphics (most in the range of $160 – $300).

Where is the circuit (inexpensive) that will accept 8 bits of x, y, brightness (whatever the TV will need) and convert it to video for my portable TV?

My 6800 is waiting, as are many others!! This circuit would be more versatile than a dedicated TV typewriter or game.

Let the computer owner make it to fit into his system to accommodate his needs. Let him decide on where the memory shall reside.

There is a need! See what you can do!

Paul Hyde Jr
Milwaukee WI

Take a look at "Build a Television Display" by C W Gantt in the June 1976 issue of BYTE, page 16. If "will accept 8 bits" is loosely interpreted as an ASCII character, then Mr Gantt's circuit fills the bill.

However, the author-engineers in our audience should take a cue from this letter of Mr Hyde. Here is a functional specification challenge:

> *Design a 256 by 256 point display module which maps 2^{16} bits of memory (organized as 2^{13} = 8 K bytes) into a standard EIA composite video signal which will drive a TV monitor.*

We'll all look forward to the first such high resolution plotting module since it will enable us to do detailed graphics for a "reasonable" price. But TANSTAAFL (There Ain't No Such Thing As A Free Lunch). The minimum price you can expect on such a graphics display is the cost of the memory and the cost of the video monitor combined.

Ask BYTE

What computer systems are available that would store five thousand (5,000) patient records? Each record would consist of name, address, telephone number, age, prescriptions and date of last examinations.

Dr Sydney B Schrum
Goldsboro NC

BYTE Replies:

A first step is to calculate specifically what your data base requirements are. Then what you want in the way of access and manipulation of the data. In your case, picking numbers which look "typical," suppose each record is (worst case lengths):

Name:		*30 bytes*
Address:	*line #1*	*30 bytes*
	line #2	*30 bytes*
	zip	*5 bytes*
Telephone:		*10 bytes*
Age:		*2 bytes*
Prescriptions text:		*30 bytes (? length)*
Date of last exam:		*6 bytes*

A fixed length record format based on this information would require 143 bytes per patient. Multiply this figure by 5,000 to find the total storage requirement of 715,000 bytes. If you use a varying length record format, this number can be shrunk considerably (perhaps 20-50%); since the typical name, address, prescription, etc, will not use all of the worst case length allowed to it.

Now, what does this mean in terms of system hardware? The implications depend upon how you want to get at the data. If you only intend to access the data serially in a personal computer version of a "batch" tape oriented system, several channels of audio cassette would be an inexpensive but slow access method. Assuming a high performance cassette interface at 1200 baud, and a 12 bit UART data format (start, 8 data bits, 1 parity bit, 2 stop bits), it will take you a total of 7150 seconds or nearly two hours to read, process and write the data in an update run. It would thus take two hours worth of cassettes or 4 standard C-30 cassettes, and you would have to change tapes every half hour.

If you want "instantaneous" access, then a higher cost floppy disk system would be required. With one floppy drive, you can store "on line" typically 200,000 to 250,000 bytes depending upon formatting. Thus you would have to spread your files out over about 3 to 4 diskettes in order to store all 715,000 records. "Instantaneous" would work out to be the time required to load and start the proper disk then access the desired patient record. If all the records are to be stored on line at one time, three to four floppy disk drives are required (or two dual disk drives).

Ancillaries you will need to make your software development convenient in either access method are a high level language like BASIC with string capability (to handle names, addresses, etc) and probably 4 K to 8 K or programmable memory work space in addition to the space required for the BASIC interpreter. You should also have a hard copy printer such as a Teletype if you want to program your patient billing and appointment reminder messages. ■

Some letters to BYTE are technical in nature and are best served by some form of response. We will try to answer as many such letters as possible. If you have a puzzle concerning some aspect of the personal computing field, write down a clear statement of your question and send it to:

Ask BYTE
Byte Publications Inc
70 Main St
Peterborough NH 03458.

We will publish names and addresses of individuals making inquiries unless you specifically request us to omit the reference. ■

BYTE reader service

To get further information on the products advertised in BYTE, fill out the reader service card with your name and address. Then circle the appropriate numbers for the advertisers you select from this list. Add a 9 cent stamp to the card, then drop it in the mail. Not only do you gain information, but our advertisers are encouraged to use the marketplace provided by BYTE. This helps us bring you a bigger BYTE.

A75 Advanced Microcomputer Products 77
A70 American Microprocessor 77
BYTE's Books 82, 84, 102
BYTE's Subscriptions 80
BYTE's Poster 96
A88 Cambion 77
A37 Celdat 91
A89 CFR Associates 73
A91 C. Itoh 83
Computer Art 78
A92 Computer Graphics 95
A81 Computers and Stuff 73
A83 Creative Computing Books 79
A87 Creative Computing Subscriptions 89
A41 Cromemco 1
A85 Custom Design Services 77
A97 Data Domain 75
A7 Delta 101
A78 Digital Group 5
A8 Dutronics 81
A9 Godbout 99
A69 HAL Communications 51
* IMS Associates, Inc CIII, 39, 112
A38 Intelligent Systems 53
A15 James 97, 105
A90 Logical Services 73
A94 Merrimac 93
A18 Meshna 110, 111
A93 Micon 73
A95 Micro Data 89
A71 Micro Peripherals 93
A57 Mikos 29
A20 Mikra-D 85
* MITS CIV, 45
A22 National Multiplex 43
A64 Oliver Audio Engineering 77
A98 On-Line 77
A63 Parasitic Engineering 85
A23 Polymorphic Systems 49
A24 Processor Tech 56, 57
A26 Scelbi 17
A27 S D Sales 108, 109
A59 Solid State Sales 103
A29 Southwest Tech CII
A30 Sphere 21
A96 Synchronous Sound 73
A32 Tri Tek 107

***Reader service inquiries not solicited. Correspond directly with company.**

7400N TTL

SN7400N*	.13	SN7453N	.27	SN74150N	1.00
SN7401N	.16	SN7454N	.20	SN74151N	.79
SN7402N	.21	SN7459A	.25	SN74153N	.89
SN7403N	.16	SN7460N	.22	SN74154N*	1.25
SN7404N*	.16	SN7470N	.45	SN74155N	.99
SN7405N	.24	SN7471N*	.69	SN74156N	.99
SN7406N	.20	SN7472N	.39	SN74157N	.99
SN7407N	.29	SN7473N*	.37	SN74160N	1.25
SN7408N	.25	SN7474N*	.32	SN74161N	.99
SN7409N	.25	SN7475N*	.50	SN74163N*	.99
SN7410N*	.16	SN7476N*	.32	SN74164N	1.10
SN7411N	.30	SN7479N*	5.00	SN74165N	1.10
SN7412N	.33	SN7480N	.50	SN74166N	1.25
SN7413N	.45	SN7482N	.98	SN74167N	5.50
SN7414N	.70	SN7483N	.70	SN74170N	2.10
SN7416N	.35	SN7485N	.89	SN74172N	18.00
SN7417N	.35	SN7486N	.39	SN74173N	1.50
SN7420N	.21	SN7488N	3.50	SN74174N	1.25
SN7421N	.33	SN7489N	2.25	SN74175N	.99
SN7422N*	.49	SN7490N*	.45	SN74176N	.90
SN7423N	.37	SN7491N	.75	SN74177N	.90
SN7425N	.29	SN7492N	.49	SN74180N	.99
SN7426N	.29	SN7493N*	.49	SN74181N	2.49
SN7427N	.37	SN7494N	.79	SN74182N	.95
SN7429N	.42	SN7495N	.79	SN74184N	1.95
SN7430N	.26	SN7496N	.89	SN74185N	2.20
SN7432N	.31	SN7497N*	4.00	SN74186N*	5.00
SN7437N	.27	SN74100N*	1.00	SN74187N	6.00
SN7438N	.27	SN74107N	.39	SN74190N	1.19
SN7439N	.25	SN74121N*	.39	SN74191N	1.25
SN7440N	.15	SN74122N	.39	SN74192N*	.89
SN7441N	.89	SN74123N*	.70	SN74193N*	.89
SN7442N	.59	SN74125N	.60	SN74194N	1.25
SN7443N	.75	SN74126N	.60	SN74195N	.75
SN7444N	.75	SN74132N	1.09	SN74196N	1.25
SN7445N	.75	SN74136N	.95	SN74197N	.75
SN7446N	.81	SN74141N	1.15	SN74198N	1.75
SN7447N*	.69	SN74142N*	4.00	SN74199N	1.75
SN7448N	.79	SN74143N*	4.50	SN74200N	5.59
SN7450N	.26	SN74144N*	4.50	SN74279N*	.90
SN7451N	.27	SN74145N	1.15	SN74251N	1.79
		SN74147N	2.35	SN74284N	6.00
		SN74148N	2.00	SN74285N	6.00

MANY OTHERS AVAILABLE ON REQUEST
20% Discount for 100 Combined 7400's

CMOS

CD4000	.25	CD4035	1.85	74C04N	.75
CD4001	.25	CD4040	2.45	74C10N	.65
CD4002	.25	CD4042	1.90	74C20N	.65
CD4006	2.50	CD4044	1.50	74C30N	.65
CD4007	.25	CD4046	2.51	74C42N	2.15
CD4009	.59	CD4047	2.75	74C73N	1.50
CD4010	.59	CD4049	.79	74C74	1.15
CD4011	.25	CD4050	.79	74C90N	3.00
CD4012	.25	CD4051	2.95	74C95N	2.00
CD4013	.47	CD4053	2.95	74C107N	1.25
CD4016	.56	CD4060	3.25	74C151	2.90
CD4017	1.35	CD4066	1.75	74C154	3.00
CD4019	.55	CD4069	.45	74C157	2.15
CD4020	1.49	CD4071	.45	74C160	3.25
CD4022	1.25	CD4072*	.45	74C161	3.25
CD4023	.25	CD4081	.45	74C163	3.00
CD4024	1.50	CD4511	2.50	74C164	3.25
CD4025	.25	CD4518	2.50	74C173	2.60
CD4027	.69	74C00N	.39	74C193	2.75
CD4028	1.65	74C02N	.55	74C195	2.75
CD4029	2.90			MC4044*	4.50
CD4030	.65			MC14016*	.56

LINEAR

LM300H	.80	LM377N	4.00	LM1351N	1.65
LM301H	.35	LM380N	1.39	LM1414N	1.75
LM301CN	.35	LM380CN	1.05	LM1458C	.65
LM302H	.75	LM381N	1.79	LM1496N	.95
LM304H	1.00	LM382N	1.79	LM1556V	1.85
LM305H	.95	NE501K	8.00	LM2111N	1.95
LM307CN	.35	NE510A	6.00	LM2901N	2.95
LM308H	1.00	NE531H	3.00	LM3065N	.69
LM308CN	1.00	NE536T	6.00	LM3900N	.55
LM309H	1.10	NE540L	6.00	LM3905N	.60
LM309K	99	NE550N	.79	LM3909*	1.25
LM310CN	1.15	NE553	2.50	LM5556N	1.85
LM311H	.90	NE555V*	.45	MC5558V	1.00
LM311N	.90	NE560B*	5.00	LM7525N	.90
LM318CN	1.50	NE561B*	5.00	LM7535N	1.25
LM319N	1.30	NE562B*	5.00	8038B*	4.95
LM319D	9.00	NE565H*	.99	LM75450	.49
LM320K-5	1.35	NE565N*	1.25	75451CN	.39
LM320K-5.2	1.35	NE566CN*	1.95	75452CN	.39
LM320K-12	1.35	NE567H*	1.25	75453CN	.39
LM320K-15	1.35	NE567V*	1.50	75454CN	.39
LM320T-5*	1.75	LM703CN	.45	75491CN	.79
LM320T-8*	1.75	LM709H	.29	75492CN	.89
LM320T-12*	1.75	LM709N	.29	75494CN	.89
LM320T-18*	1.75	LM710N	.79	**RCA LINEAR**	
LM323K-5*	9.95	LM711N	.39	CA3013	1.70
LM324N	1.80	LM723N	.55	CA3023	2.15
LM339N	1.70	LM723H	.55	CA3035	2.25
LM340K-5	1.95	LM733N	1.00	CA3039	1.35
LM340K-12	1.95	LM739N	1.29	CA3046	1.15
LM340K-15	1.95	LM741CH	.35	CA3059	2.46
LM340K-24	1.95	LM741CN	.35	CA3060	2.80
LM340T-5	1.75	LM741 14N	.39	CA3080	.85
LM340T-6	1.75	LM747H	.79	CA3081	1.75
LM340T-8*	1.75	LM747N	.79	CA3082	2.00
LM340T-18*	1.75	LM748H	.39	CA3083	1.60
LM340T-12	1.75	LM748N	.39	CA3086	.59
LM340T-15	1.75	LM1303N	.90	CA3089	3.25
LM340T-24	1.75	LM1304N	1.19	CA3091	8.25
LM350N	1.00	LM1305N	1.40	CA3123	1.85
LM351CN	.65	LM1307N	.85	CA3130*	1.49
LM370N	1.15	LM1310N	2.95	CA3600	1.75
LM370H	1.15			RG4194	5.95
LM373N	3.25			RC4195	3.25

KITS EXAR ICS

XR-2206KA **SPECIAL $17.95**
Includes monolithic function generator IC, PC board, and assembly instruction manual.

XR-2206KB **SPECIAL $27.95**
Same as XR 2206KA above and includes external components for PC board.

TIMERS		**STEREO DECODERS**	
XR-555CP	$.69	XR-1310P	$3.20
XR-320P	1.55	XR-1310EP	3.20
XR-556CP	1.85	XR1800P	3.20
XR-2556CP	3.20		
XR-2240CP	3.25	**WAVEFORM GENERATORS**	
		XR-205	8.40
PHASE LOCKED LOOPS		XR-2206CP	4.49
XR-210	5.20	XR-2207CP	3.85
XR-215	6.60	**MISCELLANEOUS**	
XR-567CP	1.95	XR-2211CP	6.70
XR-2567CP	2.99	XR-2261	3.79
		XR4136	2.00

DATA HANDBOOKS

7400	Pin-out & Description of 5400/7400 ICS	**$2.95**
CMOS	Pin-out & Description of 4000 Series ICS	**$2.95**
LINEAR	Pin-out & Functional Description	**$2.95**

ALL THREE HANDBOOKS $6.95

CONSUMER ELECTRONICS



DIGITAL WATCH

This watch is manufactured by National Semiconductor. It provides 5 functions; hours, minutes, seconds, date, A.M. indicator dot. Accuracy is assured to 5 seconds per month by precision quartz crystal. If something should go wrong with the watch, repair is assured within 48 hours after it is received. Complete with steel black leather band.

ES4-YS
3 MICRON GOLD PLATE BEZEL
$29.95
NOT A KIT



DIGITAL ALARM CLOCK

This 4 digit Novus Alarm Clock is a very reliable and smartly styled unit. It provides such features as an alarm settable to any minute of the day, a 7 minutes snooze alarm, a power failure indicator, and even an A.M., P.M. indicator.

NOT A KIT **$19.95**

OPTO ELECTRONICS

XCITON
LITRONIX
MONSANTO

DISCRETE LEDS

R - RED
G - GREEN
Y - YELLOW
O - ORANGE

.125" dia.		.185" dia.		.190" dia.	
XC209R	5/$1	XC526R	5/$1	XC111R	5/$1
XC209G	4/$1	XC526G	4/$1	XC111G	4/$1
XC209Y	4/$1	XC526Y	4/$1	XC111Y	4/$1
XC209O	4/$1	XC526O	4/$1	XC111O	4/$1

.200" dia.		.200" dia.		.085" dia.	
XC22R	5/$1	XC556R	5/$1	MV50	
XC22G	4/$1	XC556G	4/$1	.085" dia. Micro	
XC22Y	4/$1	XC556Y	4/$1	red led	
XC22O	4/$1	XC556O	4/$1	6/$1	

DISPLAY LEDS



TYPE	POLARITY	HT		TYPE	POLARITY	HT	
MAN 1	COMMON ANODE	.270	$3.25	MAN 74	COMMON CATHODE	.300	$1.50
MAN 2	5 x 7 DOT MATRIX	.300	4.95	DL707	COMMON ANODE	.300	$1.50
MAN 3	COMMON CATHODE	.125	.39	DL747	COMMON ANODE	.600	2.25
MAN 4	COMMON CATHODE	.187	1.95	DL750	COMMON CATHODE	.600	2.49
MAN 7	COMMON ANODE	.300	1.50	DL33B	COMMON CATHODE	.110	1.25
MAN 7G	COMMON ANODE-GREEN	.300	2.50	FND70	COMMON CATHODE	.250	.50
MAN 7Y	COMMON ANODE-YELLOW	.300	2.50	FND503	COMMON CATHODE	.500	1.75
MAN 72	COMMON ANODE	.300	1.50	FND507	COMMON ANODE	.500	1.75

IC SOLDERTAIL — LOW PROFILE (TIN) SOCKETS

	1-24	25-49	50-100		1-24	25-49	50-100
8 pin	$.17	.16	.15	24 pin	$.38	.37	.36
14 pin	.20	.19	.18	28 pin	.45	.44	.43
16 pin	.22	.21	.20	36 pin	.60	.59	.58
18 pin	.29	.28	.27	40 pin	.63	.62	.61
22 pin	.37	.36	.35				

SOLDERTAIL STANDARD (TIN)

14 pin	$.27	.25	.24	28 pin	$ .99	.90	.81
16 pin	.30	.27	.25	36 pin	1.39	1.26	1.15
18 pin	.35	.32	.30	40 pin	1.59	1.45	1.30
24 pin	.49	.45	.42				

SOLDERTAIL STANDARD (GOLD)

8 pin	$.30	.27	.24	24 pin	$. 70	.63	.57
14 pin	.35	.32	.29	28 pin	1.10	1.00	.90
16 pin	.38	.35	.32	36 pin	1.75	1.40	1.26
18 pin	.52	.47	.43	40 pin	1.75	1.59	1.45

WIRE WRAP SOCKETS (GOLD) LEVEL #3

10 pin	$.45	.41	.37	24 pin	$1.05	.95	.85
14 pin	.39	.38	.37	28 pin	1.40	1.25	1.10
16 pin	.43	.42	.41	36 pin	1.59	1.45	1.30
18 pin	.75	.68	.62	40 pin	1.75	1.55	1.40

50 PCS. RESISTOR ASSORTMENTS $1.75 PER ASST.

		10 OHM	12 OHM	15 OHM	18 OHM	22 OHM	
ASST. 1	**5 ea.**	27 OHM	33 OHM	39 OHM	47 OHM	56 OHM	**1/4 WATT 5% = 50 PCS.**
		68 OHM	82 OHM	100 OHM	120 OHM	150 OHM	
ASST. 2	**5 ea.**	180 OHM	220 OHM	270 OHM	330 OHM	390 OHM	**1/4 WATT 5% = 50 PCS.**
		470 OHM	560 OHM	680 OHM	820 OHM	1K	
ASST. 3	**5 ea.**	1.2K	1.5K	1.8K	2.2K	2.7K	**1/4 WATT 5% = 50 PCS.**
		3.3K	3.9K	4.7K	5.6K	6.8K	
ASST. 4	**5 ea.**	8.2K	10K	12K	15K	18K	**1/4 WATT 5% = 50 PCS.**
		22K	27K	33K	39K	47K	
ASST. 5	**5 ea.**	56K	68K	82K	100K	120K	**1/4 WATT 5% = 50 PCS.**
		150K	180K	220K	270K	330K	
ASST. 6	**5 ea.**	390K	470K	560K	680K	820K	**1/4 WATT 5% = 50 PCS.**
		1M	1.2M	1.5M	1.8M	2.2M	
ASST. 7	**5 ea.**	2.7M	3.3M	3.9M	4.7M	5.6M	**1/4 WATT 5% = 50 PCS.**

ALL OTHER RESISTORS FROM 2.2 OHMS 5.6M AVAILABLE IN MULTIPLES OF 5 ea
5-25 PCS: .05 ea. 30-95 PCS: .04 ea. 100-495 PCS: .03 ea. 500-995: .0275 ea.

14 PCS. POTENTIOMETER ASSORTMENTS

ASST. A 2 ea: 10 OHM-20 OHM-50 OHM-100 OHM-200 OHM-250 OHM-500 OHM
ASST. B 2 ea: 1K, 2K, 2.5K, 10K, 20K, 25K, 50K
ASST. C 2 ea: 50K, 100K, 200K, 250K, 500K, 1M, 2M

$9.95 Per Asst.



Each assortment contains 14 pcs of 10 turn pots. All pots are available in single unit quantities. **$.99 ea.**

Astrisk Denotes Items On Special For This Month
Satisfaction Guaranteed. $5.00 Min. Order. U.S. Funds.
California Residents — Add 6% Sales Tax — Data Sheets 25c each
Send a 13c Stamp (postage) for a FREE 1976 Catalog

JAMES

1021 HOWARD ST., SAN CARLOS, CA. 94070
PHONE ORDERS — (415) 592-8097

74LS00 TTL

74LS00	.39	74LS55	.39	74LS151	1.55
74LS02	.39	74LS73	.65	74LS153	1.89
74LS03	.39	74LS74	.65	74LS157	1.55
74LS04	.45	74LS75	.79	74LS162	2.25
74LS05	.45	74LS76	.65	74LS163	2.25
74LS08	.39	74LS83	2.19	74LS164	2.25
74LS10	.39	74LS86	.65	74LS181	3.69
74LS13	.79	74LS90	1.25	74LS190	2.85
74LS14	2.19	74LS92	1.25	74LS191	2.85
74LS20	.39	74LS93	1.25	74LS192	2.85
74LS26	.49	74LS95	2.19	74LS193	2.85
74LS27	.45	74LS96	1.89	74LS194	2.25
74LS28	.49	74LS107	.65	74LS195	2.25
74LS30	.39	74LS112	.65	74LS257	1.89
74LS32	.45	74LS132	1.55	74LS260	.55
74LS40	.49	74LS136	.65	74LS279	.79
74LS51	.39	74LS138	1.89	74LS670	5.95

DL728

The DL728 is a dual 0.5" common cathode red display. It is ideal for use with clock chips, as segments are already multiplexed. **$2.95**

MINATURE TOGGLE

DPDT				JMT	1	2-9	10-29	30-99
	ON	OFF	ON	221	3.05	2.65	1.97	1.78
	ON	NONE	ON	223	2.65	2.25	1.68	1.51
	ON	OFF	ON	121	2.35	1.95	1.43	1.30
	ON	NONE	ON	123	2.05	1.65	1.21	1.10



PUSH BUTTON

SPDT – SERIES PB, PB1

	Model Number	1 Each	2-9	10-29	30-99
Maintained Action Switch Push To Make Contact. And Push Again To Reserve Action	PB-123	$2.35	$1.95	$1.47	$1.30
Momentary Action Switch Normally Open or Normally Closed. (For N.C circuit use terminals 2 & 1; for N.O. 2 & 3; for N.C and N.O. 1,2,&3	PB-126	$2.35	$1.95	$1.47	$1.30

Quanity net prices

5 AMP RATINGS

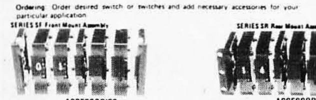


THUMBWHEEL SWITCHES

THUMBWHEEL SWITCH ONLY
SF – Front Mount SR – Rear Mount

Part No.	Description	Price
SF 12	Single Pole 10 Position	$2.50
SR 12	Decimal	3.00
SF 21	10 Position BCD only	2.50
SR 21	" "	3.00

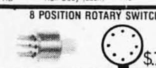
Ordering: Order desired switch or switches and add necessary accessories for your particular application.



ACCESSORIES

Part No.	Description	Price	Part No.	Description	Price
SF EP	End Plate (Pair)	$.50	SR EP	End Plates (pair)	$.50
SF DP	Divider Plate (each)	.40	SR DP	Divider Plate (each)	.40
SF BB	Blank Body (each)	.40	SR BB	Blank Body (each)	.40
SF HB	Half Body (each)	.40	SR HB	Half Body (each)	.40

8 POSITION ROTARY SWITCH



$.79

These switches are a 7 position, one position open, rotary switch enclosed in a TO-5 can. They have a standard 8 pin configuration and will mount perfectly on printed circuit board.

ZENERS—DIODES—RECTIFIERS

TYPE	VOLTS	W	PRICE	TYPE	VOLTS	W	PRICE
IN746	3.3	400m	4/1.00	1N4005*	600 PIV	1 AMP	10/1.00
1N751A	5.1	400m	4/1.00	1N4006*	800 PIV	1 AMP	10/1.00
1N752	5.6	400m	4/1.00	1N4007*	1000 PIV	1 AMP	10/1.00
1N753	6.2	400m	4/1.00	1N3600	50	200m	6/1.00
1N754	6.8	400m	4/1.00	1N4148	75	10m	15/1.00
1N965B	15	400m	4/1.00	1N4154	35	10m	12/1.00
1N5232	5.6	500m	28	1N4734	5.6	1w	28
1N5234	6.2	500m	28	1N4735	6.2	1w	28
1N5235	6.8	500m	28	1N4736	6.8	1w	28
1N5236	7.5	500m	28	1N4738	8.2	1w	28
1N456	25	40m	6/1.00	1N4742	12	1w	28
1N458	150	7m	6/1.00	1N4744	15	1w	28
IN485A	180	10m	5/1.00	1N1183	50 PIV	35 AMP	1.60
1N4001	50 PIV	1 AMP	12/1.00	1N1184	100 PIV	35 AMP	1.70
1N4002	100 PIV	1 AMP	12/1.00	IN1186	200 PIV	35 AMP	1.80
1N4003	200 PIV	1 AMP	12/1.00	1N1188	400 PIV	35 AMP	3.00
1N4004	400 PIV	1 AMP	12/1.00				

TRANSISTORS

MPS A05	5/$1.00	PN3567	3/$1.00	PN4249	4/$1.00
MPS A06	5/$1.00	PN3568	4/$1.00	PN4250	4/$1.00
2N2219A	3/$1.00	PN3569	4/$1.00	2N4400	4/$1.00
2N2221	4/$1.00	2N3704	5/$1.00	2N4401	4/$1.00
2N2222A	5/$1.00	2N3705	5/$1.00	2N4402	4/$1.00
2N2369	5/$1.00	2N3706	5/$1.00	2N4403	4/$1.00
2N2369A	4/$1.00	2N3707	5/$1.00	2N4409	5/$1.00
FN2415	5/$1.00	2N3711	5/$1.00	2N5086	4/$1.00
2N2484	4/$1.00	2N3724	$.65	2N5087	4/$1.00
2N2906A	4/$1.00	2N3725	$1.00	2N5088	4/$1.00
2N2907A	5/$1.00	2N3903	5/$1.00	2N5089	4/$1.00
2N2925	5/$1.00	2N3904	4/$1.00	2N5129	5/$1.00
2N3053	2/$1.00	2N3905	4/$1.00	2N5138	5/$1.00
2N3055	$ .89	2N3906	4/$1.00	2N5139	5/$1.00
MJE3055	$1.00	2N4013	3/$1.00	2N5209	5/$1.00
MJE2955	$1.25	2N4014	3/$1.00	2N5951	5/$1.00
2N3392	5/$1.00			C106B1SCR	2/$1.00
2N3398	5/$1.00			2N5432	$2.00

CAPACITOR CORNER

50 VOLT CERAMIC DISC CAPACITORS

	1-9	10-49	50-100		1-9	10-49	50-100
	.05	.04	.03	.001μF	.05	.04	.035
	.05	.04	.03	.0047μF	.05	.04	.035
	.05	.04	.03	.01μF	.05	.04	.035
100 pf	.05	.04	.03	.022μF	.06	.05	.04
220 pf	.05	.04	.03	.047μF	.06	.05	.04
470 pf	.05	.04	.035	.1μF	.12	.09	.075

100 VOLT MYLAR FILM CAPACITORS

.001mf	.12	.10	.07	.022mf	.13	.11	.08
.0022	.12	.10	.07	.047mf	.21	.17	.13
.0047mf	.12	.10	.07	.1mf	.27	.23	.17
.01mf	.12	.10	.07	22mf	.33	.27	.22

+ 20% DIPPED TANTALUMS (SOLID) CAPACITORS

.1 35V	.28	.23	.17	1.5 35V	.30	.26	.21
.15 35V	.28	.23	.17	2.2 25V	.31	.27	.22
.22 35V	.28	.23	.17	3.3 25V	.31	.27	.22
.33 35V	.28	.23	.17	4.7 25V	.32	.28	.23
.47 35V	.28	.23	.17	6.8 25V	.36	.31	.25
.68 35V	.28	.23	.17	10 25V	.40	.35	.29
1.0 35V	.28	.23	.17	15 25V	.63	.50	.40

MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS

Axial Lead				Radial Lead			
.47 50V	.15	.13	.10	.47 25V	.15	.13	.10
1.0 50V	.16	.14	.11	.47 50V	.16	.14	.11
3.3 50V	.15	.13	.10	1.0 16V	.15	.13	.10
4.7 25V	.16	.14	.12	1.0 25V	.16	.14	.11
10 25V	.15	.13	.10	1.0 50V	.16	.14	.11
10 50V	.16	.14	.12	4.7 16V	.15	.13	.10
22 25V	.17	.15	.12	4.7 25V	.15	.13	.10
22 50V	.24	.20	.18	4.7 50V	.16	.14	.11
47 25V	.19	.17	.15	10 16V	.14	.12	.09
47 50V	.25	.21	.19	10 25V	.15	.13	.10
100 25V	.24	.20	.18	10 50V	.16	.14	.12
100 50V	.35	.30	.28	47 50V	.24	.21	.19
220 25V	.32	.28	.25	100 16V	.19	.15	.14
220 50V	.45	.41	.38	100 25V	.24	.20	.18
470 25V	.33	.29	.27	100 50V	.35	.30	.28
1000 16V	.55	.50	.45	220 16V	.23	.17	.16
2200 16V	.70	.62	.55	470 25V	.31	.28	.26

Classified Ads for Individuals and Clubs

Readers who have equipment, software or other items to buy, sell or swap should send in a clearly typed or printed notice to that effect. The notices are free of charge and will be printed one time only on a space available basis. Insertions should be limited to no more than 100 words. Notices can be accepted from individuals or bona fide computer users' clubs only. Commercial advertisers should contact Virginia Peschke at BYTE for the latest rate card and terms.

Before sending your classified ad to BYTE, read it over. Did you include your name, address, phone number (with area code) in the text of the ad? BYTE has received several ads with incomplete phone numbers or missing addresses.

WANTED: Any low cost computer terminal for private use. Please contact Robert Brodie, 15 Harwich Rd, Chestnut Hill MA 02167, (617) 332-8034.

The Beta Terminal owners group of the Computer Hobbyist Group of North Texas is interested in establishing communications with owners of terminals that use the Univac 0769 Series print mechanism. We are looking for Beta keyboards (Microswitch No. 53SW1-2). One of our members has some spare parts for Beta terminals for sale. Contact L G Walker, Rt 1 Box 272, Aledo TX 76008, (817) 244-1013.

WANTED: Complete working computer or one still in kit form. Need E&L LR-outboards and E&L mark 80 microcomputer system or its equivalent. State condition and give lowest price. John Waskowitz, 35-30 73rd St, Flushing NY 11372.

WANTED: New or used: computers; peripherals; components; plans; software; robots; laser weapons; science fair projects; Star Trek items; rockets; electronics equipment; amateur radio; antennas; satellites; whatever you have I will make a cash offer. Please send your specifications to Joe Halligan, 3331 Altamont Dr, Wilmington DE 19810.

WANTED TO MEET: Individuals using the Univac 769 printers for hobby purposes. I have knowledge of how to use these printers and will share same. FOR SALE or TRADE: 3 serial punch devices from Teletype Corp, new, very similar to the ASR 35 punch units; take 20 mA current loop input of ASCII and require a drive motor. Jim Beistle, 3728 Wilkie Way, Ft Worth TX 76133.

FOR SALE: IBM Selectric Printer/Keyboard unit (Serial #4600785). Originally used as console for IBM 1410. Has 1407-type keyboard, mounts like 2741 in a table well. Convertible to computer console, terminal, or Selectric typewriter with proper user-supplied power supply and control logic. Needs minor repair. Sorry, no FE manuals or data sheets. Weight: approximately 40 lb. Best offer above $75 takes it. M Scherer, RR 2, Box 116, Bedford NY 10506.

1702s PROGRAMMED — I can program your 1702s for $3 or duplicate for $1.50. Changes in a duplication 10¢ each + duplicating charge. Also, I have a limited number of 4 KE PROM kits for the Altair/Imsai that hold up to 16 1702s for $65 each (1702s are not included). Please send material to be programmed in HEX (Base 16). Include return postage. John B Jay, 165 River Valley Rd NW, Atlanta GA 30328.

FOR SALE: Wurlitzer Electric Piano. Uses standard piano action to strike tuning forks with electronic pickup. Sounds like an amplified clavichord. Has controls for volume and optional vibrato; headphone and auxiliary amplifier jack. Best offer over $550. Call Carl Helmers, BYTE Publications, (603) 924-7217, or write to 70 Main St, Peterborough NH 03458.

FOR SALE OR TRADE: Friden Flexowriter Model FPC-8, 8 level code printer, keyboard, paper tape punch and paper tape reader with table. Excellent condition $500. Also disk memory, 27 track fixed head, 5454 bits per track $100, HP120A oscilloscope with 2 Tektronix probes $150. J G Hansen, PO Box 1337, Fort Davis TX 79734 (915) 426-3331 (evenings).

WANTED: Humane individual to provide back issues of BYTE from "GO" to number 5 inclusive. Will pay all costs and bribery. P L Christie, 20 James St, Adelaide, 5000, Australia. (I'm not necessarily interested in original copies, only the information therein. [BYTE won't mind if some individual makes an international good will gesture by loaning Mr Christie copies of BYTE #1 to #5 for the purpose of limited Xerox reproduction.])

FOR SALE: DEC PDP-8/S, one working, good condition, one for parts (both $2000), IBM 024 keypunch for experimenter $150. NCR card reader, new-unused, EM-D2, $100. Call or write Reg Conkling (516) 744-9475, Box 310, Shoreham NY 11786.

ALTAIR 8800 computer, fully assembled and operative, with 4 K static RAM, $575. MITS VLCT, assembled and working, with non-MITS keyboard and no case, $85. You pay shipping, I pay crating charges. James R Einolf, 12149 N Piney Lake Rd, Parker CO 80134. (303) 841-2105.

"Hints for troubleshooting and suggestions for application of the Viatron 21 Data Management Station." Write Greg Ludwig, Box 408, Rice Lake WI 54868.

Will swap my time to help you set up, program, develop system(s), or to give you training in the above, in return for using your system. Or would be interested, also, in swapping my time to secure a system of my own. Have an extensive background in systems/programming, many systems, many languages, including assembly and higher level. Please contact Toby Maki, 102 Minott Rd, Westminster MA 01473, (617) 874-5410.

Will trade comics, bicycle magazines, or pre-1950 radio magazines for microcomputer magazines or hardware. Ian MacMillan, PO Box 128, Mount Royal, Quebec, Canada H3P 3B9.

FOR SALE OR TRADE: 5 channel TTY; IBM SMS logic; CDC 2816 office typewriters with optical paper tape reader and high speed paper tape punch and controller (paper tape version of MTST). W Palya, St Joseph's College, Rensselaer IN 47978, (219) 866-7111.

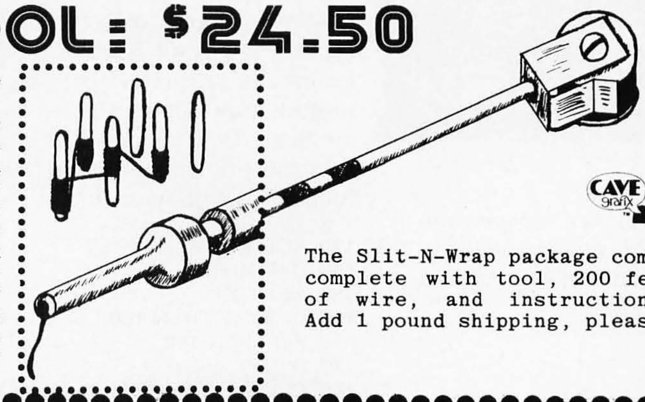
FABULOUS SALE! Spectrum Analyzer unit 10 cps to 100 kHz expand to MHz. Displays spectrum on any scope with documentation, cost $1475 new, sell for $550 or trade. VIATRON 21-PARTS! Parts, documentation and all the goodies including *RS-232* interface. Can provide repair service. R Brown, PO Box 83, Marcy NY 13403.

Bell 103 compatible Acoustic Couplers. TTL compatible data lines, single +12 volt supply. Optional signals provided: Clear to Send, DAA input. Switch selectable Up/Inverted code. We have used these with HP, DEC and Burroughs timesharing. Includes schematic and interfacing information, $50. George H Enos 2400 W Valley Pkwy, Sp 42 Escondido CA 92025.

VECTOR HAS DONE IT AGAIN !!

"SLIT-N-WRAP" TOOL: $24.50

From the same people who brought you Vectorboard, the time-saving Wire Pencil, and the Altair Universal Prototype Board comes a truly revolutionary tool. The "Slit-N-Wrap" tool provides---for the first time---a fast manual wrapping device which supplies insulated wire from a spool directly to wrap posts WITHOUT any external precutting or prestripping. It can do this in a "daisy chain", so more than one level is rarely required...thus saving space when you want to stack cards. Wires are cut off only at the end of lines---no premeasuring of wires is necessary. The secret? An almost undetectable slit is cut into the wire as it leaves the tool's tip; as the wire pulls past the wrap post, the copper is exposed to the post. Quality of wrap is extremely good---another advantage.



The Slit-N-Wrap package comes complete with tool, 200 feet of wire, and instructions. Add 1 pound shipping, please.

A FULL FEATURE ECONORAM™ 4K X 8 BOARD

$99.95

Now that other companies have picked up on our buffering and static operation...and on adding hysteresis to the data in lines (for unambiguous data transfer)...we decided to continue staying ahead of the pack by giving more features at a lower price. Like our previous memory kits, this one is FULLY PLUG-IN COMPATIBLE WITH BOTH IMSAI AND ALTAIR 8800 MICROCOMPUTERS. The board features buffered address, data out, and data in lines...zero wait states...1A typical power consumption...low power Schottky for all TTL support ICs...on board regulation with 3 regulators, situated for optimum thermal stability...lots of bypassing (39 disc ceramics, and 7 tantalums for power supply decoupling)...a DIP switch for the address select...an industrial-quality, double-sided plated through PC board (epoxy glass base like our other kits)...sockets for all ICs...plus ease of assembly, thanks to our simple-to-follow instructions. Documentation also includes a logic print of the memory. Others can incorporate the engineering, but we've got the recipe on keeping the price down.

8080 CHIP SET MICRO-MONITOR

THE 8080 IS A HOBBYIST STANDARD IC. BUT A MICROPROCESSOR BY ITSELF DOESN'T MEAN MUCH WITHOUT SOME MEMORY, AND PREFERABLY SOME KIND OF MONITOR ROUTINE SO YOU CAN AT LEAST FILL AND EXAMINE THE MEMORY. SO, HERE'S WHAT YOU GET IN OUR CHIP SET:

1 - TMS8080 MICROPROCESSOR
8 - 2102 1K STATIC RAMS
1 - 5204 4K EROM (eraseable, programable)

The cost of this chip set is $59.95. HOWEVER, BY ADDING $10 WE'LL PROGRAM YOUR 5204 WITH THE 8080 MICRO-MONITOR. THE SOFTWARE IN THE 5204 ALLOWS YOU TO EXAMINE AND FILL MEMORY, PUNCH AND READ PAPER TAPE, BRANCH TO---AND RECOVER FROM---PROGRAMS.

A COMPLETE SOFTWARE LISTING COMES WITH THE MICRO-MONITOR. THE MICRO-MONITOR IC IS ALSO AVAILABLE SEPARATELY FOR **$29.95.**

OTHER CHIP SETS

8008 CHIP SET $32.50

(1) 8008 CPU AND (8) 2102s. NO DATA OTHER THAN PINOUT INCLUDED.

PACE BASIC CHIP SET (16 BITS) $125.00

(1) PACE CPU...(4) DS3608...(1) DS0026...(1) DM8837..(2) 5204 EROMs..PLUS THE PACE DATA PACKET.

PACE DELUXE CHIP SET $195.00

(1) PACE CPU...(4) DS3608...(1) DS0026...(1) DM8837...(32) 2102s...(4) 5204 EROMs, AND OUR PACE DATA PACKET.

SENTRY CRYSTALS

$4.95 EACH. EXCELLENT FOR FREQUENCY CALIBRATORS, HAMS, OR COMPUTER BUGS. SERIES MODE, FUNDAMENTAL OPERATION, WIRE LEADS. CHOOSE FROM THE FOLLOWING FREQUENCIES:

4 MHz (PACE CLOCK FREQUENCY), 5 MHz, 8 MHz, 10 MHz, 12 MHz, 15 MHz, 18 MHz (8080 CLOCK FREQUENCY), AND 20 MHz.

FREE FLYER!

The best thing in life is free, so they say...but so are some of the second and third best things, like our flyer. Send us your name and address if you're in a hurry, or circle the reader service card if you can afford to wait a little longer.

TERMS: Add 50¢ to orders under $10. We pay the freight unless otherwise indicated for an item. Cal res add sales tax. No COD orders---it's too much paperwork. Bankamericard®/Mastercharge®: call (415) 357-7007.

GODBOUT

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

BYTE'S BUGS

Here lies documentation of known bugs detected in previous editions of BYTE . . .

Patching the Biorhythm Program

Don Libes, 995 Chimney Ridge, Springfield NJ, reports that the biorhythm program in the April issue of BYTE (page 22) incorrectly calculates a person's life span. He suggests the following patch to correct the problem. Delete lines 130 to 470 of the listing as printed, and replace them with the following set of lines:

```
130 FOR X=1, M–1
140 D=T(X)+D
150 NEXT X
160 IF M<=2 THEN 180   or IF Y/4-INT(Y/4)
      ≠ 0 THEN 180
170 D=D+1
180 D=(Y–1)*365+INT((Y–1)/4)+D
190 D4=D1
200 FOR X=1,M1–1
210 D4=T(X) + D4
220 NEXT X
230 IF M1=<2 THEN 250   or IF Y/4–INT(Y/4)
      ≠ 0 THEN 250
240 D4=D4+1
250 D4=(Y1–1)*365+INT((Y1–1)/4)+D4
260 D3=D4–D
```

With this patch, all cycles should now start in phase at one's birthdate and calculate the biorhythm correctly according to the biorhythm hypothesis as stated. However, this patch does not affect the question of whether the biorhythm hypothesis as stated and programmed is valid.■

How to Do It Better

Michael M Dodd, 291 Waples Estates, Fairfax VA 22030, sent us a lengthy note commenting upon the design of Robert Bosen's interface, page 42 of April BYTE. It is reprinted here with a couple of editorial notes interjected. The key to the notes is as follows: [Oops] *indicates a design flaw that should have been corrected editorially;* [Point] *indicates a possible better way to have done something;* [Good point] *indicates a place where an engineer with considerable experience would have done things differently, and the reasons why. Michael's letter is printed as much for the two real* [Oops] *class errors as it is for the tutorial value of his comments concerning design practice.*

In reading the article, "Controlling External Devices With Hobbyist Computers" by Robert Bosen in the April issue of BYTE, I noticed several cases of what I consider to be poor hardware design. I would like to point them out to possibly help someone wanting to build a similar circuit.

1. The IC inverting bit D4 in figure 1 should be a 7404, not a 7400 (the pin numbers are wrong for a 7400). *[Oops]* In addition, it is generally not good practice to parallel TTL gates; if you need more drive, split the load. It is possible that one of the gates could produce a slightly lower zero voltage and take all the current while the other one did nothing. *[Good point]*

2. Pin 10 (the preset input) and pin 4 (same) of the 7474 flip flops should be tied to +5 V; leaving it open is inviting false triggering due to noise pickup. *[Good point]*

3. Driving the base of transistors with TTL outputs can cause overheating of the device. A three volt drop across a 1 k resistor causes 3 milliamps to be drawn from the TTL output in the high state; this is far in excess of the 400 microamps specified. *[Good point]*

4. The jumpering scheme in figure 2 is wrong. It will cause the outputs of the inverters to go low when the inputs are left floating; this will short the address bus lines to ground. The jumpers should be between the inputs of the 7430 and either the outputs of the inverters or the address bus directly. *[Oops]*

5. Mr Bosen's point about saving address loading by using low power TTL is well taken. The second input of any 7400 used as an inverter should be tied to +5 V to reduce loading, not to the active input. This will save one input load on the address line. *[Good point]*

6. In figure 3a the diode across the relay coil does not have to be a Zener. When the magnetic field collapses, a spike is indeed generated, but it is of the opposite polarity; thus, a common silicon diode (polarity as shown) will shunt the spike, protecting the transistor. *[Good point]*

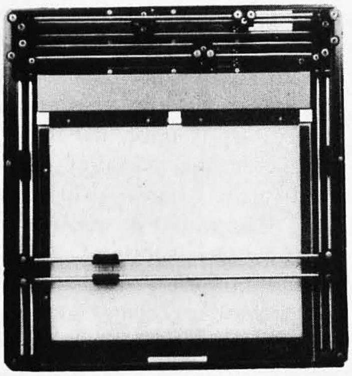
7. On the software side, I might point out that with the "clear" line on bit D4 (figure 1) connected as shown, a zero on D4 will cause a "clear" condition. Many people like to work with active high (a one performs the operation) and one inverter is required to achieve this. It sometimes helps in programming if you use active high. *[Point]*

8. Also on software, the description about clocking another output channel to get the pulse necessary to toggle the flip flop is not really necessary. The chip enable inputs to the 75154 (figure 1) are designed to put all the outputs in a high state when the chip is disabled. All that is necessary to get a pulse is to tie either pin 18 or 19 to an unused bit on the output port (there are three left) and wiggle that bit while maintaining the channel selection number on the lower four bits. This may or may not be easier than selecting another channel, but at least you don't have to worry about what channel to select to get your pulse. *[Point]*

I just wanted to point out these items. Since Mr Bosen has obviously built the device, I can't argue with success; but some errors must have crept in between his wire wrap tool and your pages.■

DELTA ELECTRONICS

POST OFFICE BOX 2, AMESBURY, MASSACHUSETTS 01913 Phone (617) 388-4705



Bolt, Baranek & Newman DATA CODER

These DATA CODERS are moving fast, but we still have a few left. They are made by Bolt, Baranek & Newman for use in the medical electronics field, but have many other applications. Any data to be digitized (charts, maps, waveforms, drawings, game grids, etc.) is placed in the bed, and the sight is moved along the data. This generates vertical & horizontal displacement codes, which can be stored and processed by your calculator or micro-computer.

The bed size is 12¾" x 10 5/8", and is translucent for use as a light table. Some units have a 7" x 10½" translucent portion, the rest being opaque. Each is divided into 128 increments, for a total of 16,384 bits over the bed. Horiz. increment size = .10", vert. increment size = .083". Each axis generates an 8 bit code by means of high reliability spring wipers on a gold plated PC encoding board. Easily interfaced to any logic family with appropriate pull-up resistors. Overall size 18¾" x 17¼" x 3" high, shipping weight 10 lbs. Complete with coding data.

Buy one while you can. It is unlikely that this item will ever become available again.

STOCK NUMBER B5352 **$79.95 each**

TRANSFORMERS

We carry one of the largest & most diversified stocks of transformers in the country. Listed below is a sampling of our many different types. Our free catalog lists many more.

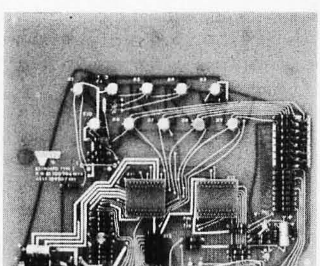
PRIMARY	SECONDARY NO. 1	NO. 2	NO. 3	NO. 4	SIZE	LBS.	STK. NO.	PRICE
115v or 230v	34v, 3 A, CT	17v, 4 amps	11v, 5 amps	6.3v, 1.5 amps	3¾ x 4½ x 4¾	10	B9397	$12.95 ea, 2/24.00
115v	26v, 1 A, CT	6.3v, ½ amp			3 x 2½ x 2¼	4	B9907	3.75 ea, 2/7.00
115v	34v, 6.5 A, CT	6v, 2 amps			3 x 3¾ x 4½	9	B9888	10.95 ea, 2/21.00
115v or 230v	17v, 4 A, CT	170v, 350 ma. (Nixie supply)			3¼ x 3¾ x3½	6	B9796	3.50 ea, 3/9.00
115v	64v, 4 amp, CT, taps for 32v, 5.5 A	17v, 7 A, CT	380v, .4 A, CT		4½ x 4 x 4	11	B9905	11.95 ea, 2/22.00
115v	18v, 4 amps	17v, 2.5 amps	10v, 5 amps		2¾ x 3¾ x 3½	6	B9906	8.95 ea, 2/16.00

AUTOTRANSFORMERS: 2 KVA rating, voltage taps every 5 volts from 85v to 125v. Valuable in areas of low or fluctuating line voltage. 4½ x 5½ x 4½, 15 lbs. STOCK NO. B9721 $9.95 ea, 2/18.00

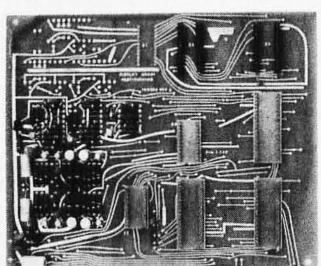
2 KVA rating, step up/step down autotransformer. Taps at 100v, 115, 125, 200, 220, and 230v. 3¾ x 4½ x 4½, 11 lbs. STOCK NO. B9856 $12.95 ea, 2/25.00

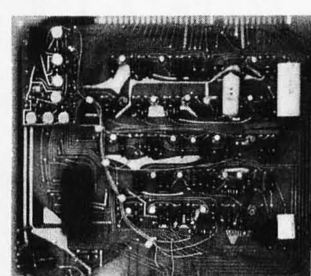
ISOLATION TRANSFORMER: 115v to 115v, 100 watts continuous. 3 x 3½ x 3½, 7 lbs. STK NO. B9971 $5.95 ea, 2/11.00

Viatron System 21 PC BOARDS



B6362





B6364

Recently, Viatron System 21 Data Management Stations have become available from several surplus sources. While this is an excellent machine, the availability of spare parts is practically nil. We have these three PC boards available for this system. They are in new, unused condition, but do NOT have the LSI chips. We are offering them for their use as replacements or for parts.

KEYBOARD TYPE 2, number 109207. STOCK NO. B6362 $9.95 each

DISPLAY ARRAY PARTITIONING, number 109300 Rev. B STOCK NO. B6363 $9.95 each

COMMUNICATIONS ADAPTOR BOARD, number 1099 Rev. B. Used in the 6003 and 6004 Communication Adapter, which provides an ASCII interface to other equipment. STOCK NO. B6364 $14.95 each

Philbrick/Nexus OPERATIONAL AMPLIFIER



Current factory price is $12.

Philbrick/Nexus Model SQ-10a is a high gain, high stability general purpose op amp. It operates over the temp. range of −25°C to +85°C. Both inputs & output are fully protected against shorts to ground or the power supply. It may be used open loop as a voltage crossing detector. 1 Mhz unity gain bandwidth. Open loop gain 60,000 min. Slew rate 1.1 V/ sec. min. 100 nA. max input bias current. Nominal power supply ±15 v. 1 1/8" x 1 1/8" x ½"

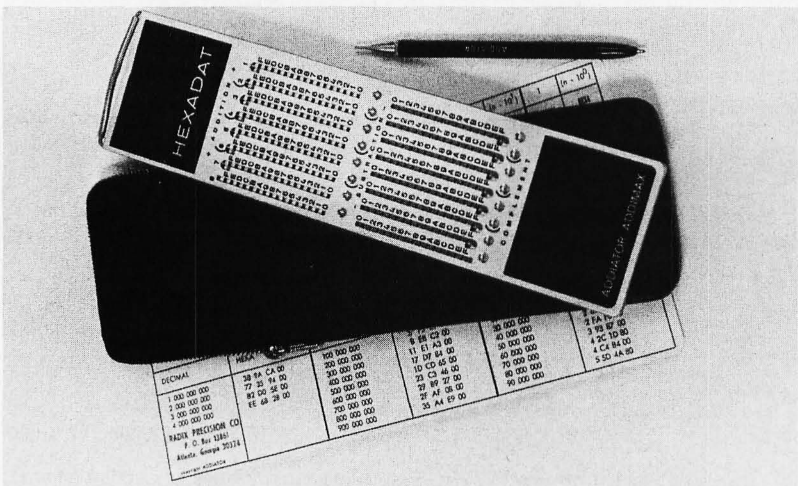
STOCK NO. B4539 **With data** **$1.75 each, 4/6.00**

Send for our latest free catalog. Minimum order $5, phone orders welcome. Include sufficient postage (2 lbs min.), excess will be refunded. BANKAMERICARD & MASTERCHARGE welcome, ALL numbers needed for processing. Minimum charge $15.

What's New?

But It's New to Me?

Yours truly [CH] had a little argument with photographer Ed Crabtree over this item, the "Hexadat" mechanical calculator available from Radix Precision Co, Atlanta GA. The "Hexadat" is a boon to all lovers of 4 bit representations of binary data as hexadecimal digits. Hexadecimal notation is the natural way to represent memory contents in lieu of other formatting conventions for any machine with a word length that is a multiple of 4 bits (but not divisible by 3 or else octal lovers will have a good argument).



But hexadecimal arithmetic is another matter, a technique which takes practice to make perfect or reliable as anyone with experience puzzling out an OS 360 or 370 core dump can attest. So, several years ago, the Radix Precision people created this neat little mechanical adding machine which is operated by a stylus and uses base 16 notation. Now that the microprocessor era of 8 bit processors is upon us, the same problem of hexadecimal addition and subtraction which afflicts users of large scale IBM machines is repeated in miniature for those of us who use hexadecimal for 8 bit processors. And this same mechanical calculator will provide excellent service for hexadecimal arithmetic in your own computer laboratory. It comes in a carrying case with a reference card for base conversions and the stylus used to perform operations.

Which brings the subject back to a certain argument. When Ed Crabtree returned with the pictures, he protested that "there's nothing new here!" Whereupon he produced from his pocket a neat little mechanical calculator circa 1900 which had been used by his grandfather. In "the old days" people apparently got along without the convenience of the $8.95 hand cal-

Continued on page 112



BOOK REVIEWS

Microprocessors and Microcomputers *by Branko Soucek. Published by John Wiley and Sons, New York.*

This over 600 page book contains important information for the experienced hardware and software person but is still very readable for the neophyte. The information presented is both valuable in learning the basics of microprocessors and in selecting a microprocessor or microcomputer.

The first 100 pages cover number systems, digital circuits, and basic microcomputer instructions in enough depth to get the beginner up to speed. In the chapter on basic microcomputer instructions and in much of the material which follows, the DEC PDP-8 minicomputer (or Intersil 6100 microprocessor) instruction set is used as an example. Although this is a somewhat primitive instruction set, it is sophisticated enough to get the concepts across without confusing the beginner with a complicated machine language.

This introduction to basic hardware is followed by about 50 pages of discussion of programming, both in assembly language and in FORTRAN. Throughout this chapter FORTRAN source code is presented with the assembly code which performs the same function.

The last chapter in Part I presents information on microcomputer IO and interfacing. The necessary hardware and software for both programmed IO and direct memory access are discussed.

Part II is a detailed discussion (almost 250 pages) of representative microprocessors. The architecture, instruction set, timing, and interfacing are discussed and programming examples are presented for the following microprocessors:

4004/4040
8008/8080 and MCOM-8
M6800
PPS-4
PPS-8
IMP 4/8/16 and PACE

Part III, called "New Microprocessors and Special-Purpose Microsystems," is much like Part II. It presents detailed discussions of microprocessors and hardware which is similar to microprocessors. The subjects include:

PDP-11 Minicomputer and
LSI-11 Microcomputer
F8 Microprocessor
SMS Microcontroller
IM6100 Microprocessor and
PDP-8 Minicomputer

Phil Hughes
PO Box 43
Richland WA 99352 ■

Amateur Radio Publications

The world of amateur radio has been one of the long time homes for individuals practicing state of the art electronics techniques. In the present state of the art, the combination of microprocessors with amateur radio communications practices leads to some exciting applications such as packet switching communications networks, automated amateur radio stations, etc. (The first prize winner in the World Altair Computer Convention which MITS put on in March was an amateur radio operator.) To provide a pointer into amateur radio circles, the following is a set of short reviews of some amateur radio publications which appear monthly.

QST

QST was the first amateur radio magazine ever. It is published monthly by the American Radio Relay League, 225 Main St, Newington CT 06111. It is professionally edited and currently is run in a full size magazine format (before January 1976, all the ham magazines were in small magazine format). Each issue typically has a mixture of technical articles, tutorial articles, and a

CRYSTALS

THESE FREQUENCIES ONLY

Part #	Frequency	Case/Style	Price
CY1A	1.000 MHz	HC33/U	$4.95
CY2A	2.000 MHz	HC33/U	$4.95
CY3A	4.000 MHz	HC18/U	$4.95
CY7A	5.000 MHz	HC18/U	$4.95
CY12A	10.000 MHz	HC18/U	$4.95
CY14A	14.31818 MHz	HC18U	$4.95
CY19A	18.000 MHz	HC18/U	$4.95
CY22A	20.000 MHz	HC18/U	$4.95
CY30B	32.000 MHz	HC18/U	$4.95

CLOCK CHIPS — CALCULATOR CHIPS

MM5309	6 Digit, BCD Outputs, Reset PIN.	$5.95
MM5311	6 Digit, BCD Outputs, 12 or 24 Hour	4.95
MM5312	4 Digit, BCD Outputs, 1 PPS Output	4.95
MM5313	6 Digit, BCD Outputs, 1PPS Output	4.95
MM5314	6 Digit, 12 or 24 Hour, 50 or 60 Hz	4.95
MM5316	4 Digit, Alarm, 1PPS Output	6.95
MM5318	Video Clock Chip, For Use With MM5841	9.95
CT7001	6 Digit, Calander, Alarm, 12 or 24 Hour	6.95
CALCULATOR CHIPS		
MM5725	6 Digit, Four Function, Less Decimal	$2.95
MM5738	8 Digit, 5 Function: +, =, ×, ÷, %	2.95
MM5739	8 Digit, 4 Function, Floating Decimal	2.95
CT5001	12 Digit, 4 Function	3.95
CT5005	12 Digit, 4 Function with Memory	5.95
CT5030	12 Digit, 4 Function and %	7.95
MISC. MOS		
MM5320	TV Camera Sync. Generator	$19.95
MM5330	4½ Digit DVM Chip	9.95
MM5369	60 Hz Timebase Circuit From 3.58 MHz	5.00
MM5841	Video Generator For MM5318	18.00
MC1408-L7	7 Bit Digital to Analog Converter	9.95
MK5007	4 Decade Counter with Latches	10.95
LD110/LD111	3½ Digit DVM Chip Set	25.00
95H90	100 MHz ÷ 10 Counter For Prescalers	13.95

Filament Transformers

FILAMENT/single secondary/50-60 Hz

Type No.	Secondary Volts	Secondary Amps	Primary Volts	RMS Test Voltage	Case Type	Connections or Lead Holes Used	Case Dimension H	Case Dimension W	Case Dimension D	Mounting Dimension MW	Mounting Dimension MD	Mtg. Hole Size	Price
F-7X	5 CT	3	115	1500	X	Leads	1-15/16	3-5/16	2	2-13/16		3/16	5.10
F-13X	6.3	6	115	1500	X	Leads	1-3/8	2-3/8	1-3/8	2		3/16	3.80
F-14X#	6.3 CT	1.2	115	Pri. 1500 Sec. 2500	X	Leads	1-5/8	2-13/16	1-5/8	2-3/8		3/16	4.15
F-16X	6.3 CT	3	115	Pri. 1500 Sec. 2500	X	Leads	1-15/16	3-5/16	2	2-13/16		3/16	5.35
F-25X	12.6 CT	1.5	115	1500	X	Leads	1-15/16	3-5/16	2	2-13/16		3/16	5.25
F-40X	26.8 CT	1	115	1500	X	Leads	2	3 1/4	2 1/8	2-13/16		3/16	7.00
F-45X	24.0 CT	1	115	1500	X	Leads	2	3 1/4	2 1/8	2 13/16		3/16	7.50
F-54X	35.0 CT	1.5	115	1500	X	Leads	2-19/32	4	2 1/4	3 9/16		3/16	8.50

Add 50¢ Postage

VECTOR WIRING PENCIL



Vector Wiring Pencil P173 consists of a hand held featherweight (under one ounce) tool which is used to guide and wrap insulated wire, fed off a self-contained replaceable bobbin, onto component leads or terminals installed on pre-punched "P" Pattern Vectorbord®. Connections between the wrapped wire and component leads, pads or terminals are made by soldering. Complete with 250 FT of red wire. **$9.50**

REPLACEMENT WIRE — BOBBINS FOR WIRING PENCIL

W36-3-A-Pkg. 3	250 ft. 36 AWG GREEN	$2.40
W36-3-B-Pkg. 3	250 ft. 36 AWG RED	$2.40
W36-3-C-Pkg. 3	250 ft. 36 AWG CLEAR	$2.40
W36-3-D-Pkg. 3	250 ft. 36 AWG BLUE	$2.40

1/16 VECTOR BOARD

0.1" Hole Spacing	Part No.	P-Pattern L	P-Pattern W	Price 1-19	Price 20-49
PHENOLIC	64P44 062XXXP	4.50	6.50	1.72	1.54
	169P44 02XXXP	4.50	17.00	3.69	3.32
EPOXY GLASS	64P44 062	4.50	6.50	2.07	1.86
	84P44 062	4.50	8.50	2.56	2.31
	169P44 062	4.50	17.00	5.04	4.53
	169P84 062	8.50	17.00	9.23	8.26
EPOXY GLASS COPPER CLAD	169P44 062C1	4.50	17.00	6.80	6.12

VECTOR TERMINALS

Series	Part No. Finish Gold	Part No. Finish Tinned	Post Size	25 pcs Gold	25 pcs Tin	50 pcs Gold	50 pcs Tin
T44	T44-1	T44	.025" sq.	1.50 lot	1.00 lot	2.75 lot	1.75 lot
T46	T46-1	T46	.028" sq.	1.75 lot	1.25 lot	3.00 lot	2.25 lot
T49	T49A	T49	.025" sq.	1.75 lot	1.25 lot	3.10 lot	2.25 lot
T42-1	PKG. 100	Terminals	$3.50 C	$13.00M			

DIP SWITCH

These switches feature seven SPST slide switches in a molded dip. They are ideally suited for microprocessor applications. **$1.95**

100 PER STRIP **MOLEX PINS**

Intended for use as an inexpensive substitute for IC sockets. Also perfect for use as board connectors and in subassemblies.

SPECIAL — 100/1.49 — 1000/12.00

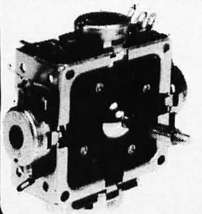
64 KEY KEYBOARD



The ideal item for the real homebrew computer hobbyist. This keyboard features 64 unencoded SPST keys, unattached to any kind of P.C.B. A very solid molded plastic 13" x 4" base suits most applications.

$24.95 ea.

JOYSTICK



These joysticks feature four 100K potentiometers, that vary resistance proportional to the angle of the stick. Sturdy metal construction with plastics components only at the movable joint. Perfect for electronic games and instrumentation.

$9.95 ea.

MICROPROCESSOR COMPONENTS

8080 SUPPORT DEVICES

8080A $37.95

8212	8 BIT INPUT/OUTPUT PORT FOR 8080	$ 5.95
8224	CLOCK GENERATOR AND DRIVER FOR 8080	12.95
8228	SYSTEM CONTROLLER AND BUS DRIVER FOR 8080	12.95

8080 $24.95

CPU'S			**RAM'S**			
8008	8 BIT CPU	$19.95	3601	256 x 1	FAST	5.00
8080	Super 8008	24.95	1101	256 x 1	Static	$ 2.25
8080A	Super 8008	37.95	1103	102 x 1	Dynamic	2.95
SR'S			2101	256 x 1	Static	6.95
2504	1024 Dynamic	$ 9.00	2102	1024 x 1	Static	1.95
2518	Hex 32 BIT	7.00	2107	4096 x 1	Dynamic	19.95
2519	Hex 40 BIT	4.00	2111	256 x 4	Static	7.95
2524	512 Dynamic	2.95	7010	1024 x 1	MNOS	29.95
2525	1024 Dynamic	6.00	7489	16 x 4	Static	2.49
2527	Dual 256 BIT	3.95	8101	256 x 4	Static	7.95
2529	Dual 512 BIT	4.00	8111	256 x 4	Static	7.95
2532	Quad 80 BIT	3.95	8599	16 x 4	Static	3.49
2533	1024 Static	7.95	91L02	1024 x 1	Static	2.49
3341	Fifo	6.95	74200	256 x 1	Static	6.95
74LS670	16 x 4 Reg.	3.95	93410	256 x 1	Static	1.75
UART'S			**PROMS**			
AY-5-1013	30K Baud	$ 6.95	1702A	2048	Famos	$15.95
ROM'S			5203	2048	Famos	14.95
2513	Char. Gen.	$11.00	82S23	32 x 8	Open C.	5.00
7488	Random Bits	3.50	82S123	32 x 8	Tristate	5.00
2516	Char. Gen.	13.00	74S287	1024	Static	7.95

2K RAM SPECIAL

MM5262 Fully decoded 2Kx1 dynamic RAM. All inputs except clocks are TTL compatible. Provides a 635 ns minimum access time, and requires, a +5, +8.5, and −15V power supply. Low power provides non volatile memory using battery back up.

$0.99 ea. (0.05 cents per bit)

Special Requested Items

RC4194	Dual Track V Reg	$ 5.95	N8T97	$ 2.00	MK5007	$10.95	MC4044	$ 4.50
RC4195	±15V Track Reg	3.25	4024P	2.25	8263	5.95	LM3909	1.25
F9368	Decoder	3.95	DM8130	3.25	8267	2.75	MM5320	19.95
LD110/111	DVM Chip Set	25.00	9322	1.30	8288	1.15	4072AE	.45
CA3130	Super CMOS Op Amp	1.49	MC14016	.56	8826	3.00	7422	1.50
MC1408L7	D/A	9.95	2525	6.00	8880	.80	7497	4.00
F3341	FIFO	6.95	2527	3.95	4511AE	2.50	74186	5.00
MM5841	Character Gen.	18.00	CD4518	2.50	XR4136	2.00	74279	.90
AY5-9100	Push Button Dialer	17.50	MM5309	5.95	4566AE	3.00		

WE'LL BE HAPPY TO QUOTE ON YOUR SPECIAL PARTS —

Continental Specialties

PROTO POWER! POWER FOR THE PROFESSIONAL



Made in USA Patents Pending

$75 Proto Board 203

Breadboard Prototesting with 5 Volt, 1 AMP Regulated Power Supply Included!

A total ready-to-use power breadboard prototest device with a built-in regulated, short-proof power supply. Just plug-in and start building! 2 extra floating 5-way binding posts for external signals. Self-contained with power switch indicator lamp and power fuse. 24-14 pin DIP cpacity. Attractive two-tone quality case. All metal construction. 9¾" L x 6½" W x 2¾" H. 5 lbs. Order today!

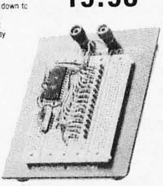
For the economy-minded experimenter . . .

NEW

Proto Board 100

19.95

Here's a low cost, big 10 IC capacity breadboard kit with all the quality of QT Sockets and the best of the Proto-Board series . . . complete down to the last nut, bolt and screw. Includes 2 QT-35S Sockets; 1 QT-35B Bus Strip; 2 5-way binding posts; 4 rubber feet, screws, nuts, bolts; and easy assembly instructions.

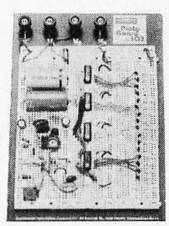


THE MINI-BREADBOARD BUDGET KIT

Proto Board 101. Compact. Inexpensive. 10-14 pin DIP capacity. 5.8" long x 4.5" wide. **29.95**

Proto Board 102. Compact 12-14 pin DIP capacity. 7" 4½". **39.95**

Proto Board 103. 2,250 solderless tie points. (4) 5-way binding posts. 24-14 pin DIP capacity. 9" x 6". **59.95**



Proto Board 104. Huge. 3,060 solderless tie points. (4) 5-way binding posts. 32-14 pin DIP capacity. 9½" long x 8" wide **79.95**

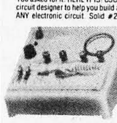
COMPLETE KIT . . .

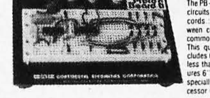
DESIGN MATE 1

CIRCUIT DESIGNER

You asked for it. HERE IT IS! CSC's basic circuit designer to help you build and test ANY electronic circuit. Solid #22 AWG wire interconnects any discreet component without solder. Resistors. Transistors. Linear/Digital ICs in TO-5 DIP packages (from 8-40 pins). And more. Components fit directly into Sockets and Bus strips instantly and easily. Plus, DESIGN MATE ONE's (DM-1) variable regulated power supply gives 5-15VDC up to 600ma (9 watts). Now that's design flexibility for you!

You can even monitor the DM-1's internal power supply or external circuits thanks to the built-in 0-15V voltmeter. If you're into electronics, DM-1 is a must for your laboratory. It comes to you completely wired, tested and ready to use for only **$49.95**





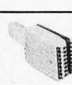
PROTO BOARD 6

The PB-6 lets the user test and build circuits without soldering or patch cords: all interconnections between components are made with common #22 AWG hook-up wire. This quality breadboarding kit includes 630 component tie points at less than 2.5 cents each. It measures 6" long by 4" wide. Designed specially to Breadboard Microporcessor Circuits. **$15.95**

Now trace signals or troubleshoot fast. Inject signals or wire unused circuits into existing boards. Scope probes, test leads lock onto Dynagrip inset. Plastic construction — no springs or pivots. Thousands of operations. Narrow throat for high density PC boards.

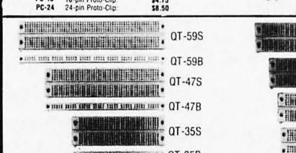
PC-14	14-pin Proto-Clip	$4.50 ea.
PC-16	16-pin Proto-Clip	$4.75
PC-24	24-pin Proto-Clip	$8.50





LOGIC MONITOR

Simultaneously displays static and dynamic logic states of DTL, TTL, HTL or CMOS DIP ICs. Pocket size. **$84.95.**



QT Proto Strips

QT type	#holes	price
QT-59S	590	12.50
QT-59B	bus strip	2.50
QT-47S	470	10.00
QT-47B	bus strip	2.25
QT-35S	350	8.50
QT-35B	bus strip	2.00
QT-18S	180	4.75
QT-12S	120	3.75
QT-8S	80	3.25
QT-7S	70	3.00

DIGITAL CLOCK KIT — 3½ INCH DIGITS



This clock features big 3½" high digits for viewing in offices, auditoriums, etc. Each digit is formed by 31 bright 0.2" LED's. The clock operates from 117 VAC, has either 12 or 24 hr. operation. The 6 digit version is 27" x 3½" x 1½", and the 4 digit is 18" x 3½" x 1½". Kits come complete with all components, case and transformer.

Specify 12 Or 24 Hr. When Ordering

4 DIGIT KIT $49.95 — 4 DIGIT ASSEMBLED $59.95

6 DIGIT KIT $69.95 — 6 DIGIT ASSEMBLED $79.95

Satisfaction Guaranteed. $5.00 Min. Order. U.S. Funds.
California Residents — Add 6% Sales Tax — Data Sheets 25c each
Send a 13c Stamp (postage) for a FREE 1976 Catalog

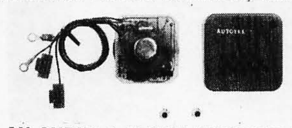
JAMES

1021 HOWARD ST., SAN CARLOS, CA. 94070

PHONE ORDERS — (415) 592-8097

AUTOTEL ™

WARNING LIGHTS ARE NOT ENOUGH



AN AUDIBLE ALARM INDICATING POTENTIAL ENGINE DAMAGE

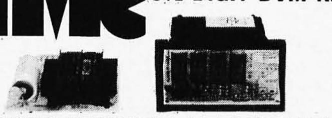
AUTOTEL is an effecient (15 ma current standby) device by which every owner of an automobile, truck or vehicle equipped with indicator lights for temperature and oil pressure can be assured of a reliable warning **before** an impending failure.

AUTOTEL, by means of an audible signal 70 db pulsing) immediately forewarns the vehicle operator of a malfunction or failure, allowing time to correct and prevent major engine failures. It is programmed so there is no sound during normal operating conditions.

AUTOTEL features CMOS circuitry, packaged in a 2½" sq. x ⅞" case. The kit comes complete with all components, hardware and case to hook directly into your car's warning light system.

$14.95 Assembled $9.95 Per Kit

iMC 3½ DIGIT DVM KIT



This 0-2 VDC .05 per cent digital voltmeter features the Motorola 3½ digit DVM chip set. It has a .4" LED display and operates from a single +5V power supply. The unit is provided complete with an injection molded black plastic case complete with Bezel. An optional power supply is available which fits into the same case as the 0-2V DVM allowing 117 VAC operation.

A. 0-2V DVM with Case $49.95

B. 5V Power Supply $14.95

JE700 CLOCK



12 or 24 Hour

115 VAC

The JE700 is a low cost digital clock, but is a very high quality unit. The unit features a simulated walnut case with dimensions of 6" x 2½" x 1". It utilizes a MAN72 high brightness readout, and the MM5314 clock chip.

$19.95

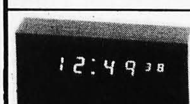
Liquid Crystal Temperature Display

Six Digit Light Emitting Diode Display

DL 33 Display



This clock makes an attractive addition to any desk. It has an extruded, black anodized aluminum case. It displays hours, minutes, and seconds with .11 inch displays, and comes complete with a liquid crystal thermometer. It operates off 115 VAC at 50 or 60 Hz. **$24.95**

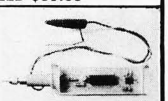


This large digit clock (.6" hours & minutes, .3" seconds) features the MM5314 clock chip. It operates from 117 VAC, and will operate in either a 12 or 24 hour mode. The clock is complete with a walnut grain case, and has fast set, slow set, and hold time set features.

KIT - ALL COMPONENTS & CASE $34.95

WIRED & ASSEMBLED $39.95

JE803 PROBE



The Logic Probe is a unit which is for the most part indespensible in trouble shooting logic families: TTL, DTL, RTL, CMOS. It derives the power it needs to operate directly off of the circuit under test, drawing a scant 10 mA max. It uses a MAN3 readout to indicate any of the following states by these symbols: (H) - 1 (LOW) - o (PULSE) - P. The Probe can detect high frequency pulses to 45 MHz. It can't be used at MOS levels or circuit damage will result.

$9.95 Per Kit printed circuit board

ELECTRONIC TOOLS

QUICK CHARGE CORDLESS SOLDERING IRON



This versatile tool weighs only six ounces, and can be operated without power anywhere. It gives the performance of a 50 watt iron with a tip temperature of over 700°F. It will solder an average of 160 3 twist 22 guage wire joints on one charge. It has a charge time of 4 hours, and heats in 5 seconds. Complete with recharger.

$29.95 EACH

NIBBLING TOOL



This Nibbling Tool is perfect for cutting, trimming or notching sheet metal up to 18 guage.

It operates like a punch and die and also works well on aluminum or plastic up to 1/16" thick. We feel that this tool can be a real time-saver for our customers when working on chassis, printed circuit boards and prototype motel parts.

Nibbling Tool $6.95
Replacement Punch $3.75 Each

DIAGONAL CUTTER

½" Semi-Flush Cut Fine Diagonal



4-in. long. Most popular cutter on assembly line-cuts at extreme tip. Semi-flush cut. For fine wire work only. No 20 soft copper wire and smaller. Width across joint 13/32" thru joint 7/32 tip length 1/2

Light Blue Handle A97MS $8.50 ea.

SOLDER



Part No.	Wire Dia.	Price/1 lb Spool
SN60	.031	6.85/Spool
SN60	.040	6.50/Spool

Solder Wick



STOCK	WIDTH	PRICE
40-1	.025	$2.00 ea
40-2	.050	2.00 ea
40-3	.075	2.00 ea



large amount of information about amateur radio happenings. As an example of technical content, the April 1976 issue featured a cover article entitled "One KW – Solid State Style, Part 1" which shows how H O Granberg, WB2BHX/OH2ZE/7 built a high power radio frequency amplifier using solid state electronics exclusively. Other April articles included the fourth part of a series on "Learning to Work With Integrated Circuits," and articles on several technical aspects of amateur radio. A major portion of the magazine is devoted to general interest items. The April issue also reported amateurs' emergency work in the Guatemalan earthquakes, and operating activities such as message networks, civil defense organizations, etc. *QST* is also chock full of advertising about amateur radio products. This journal is a "must" for radio amateurs, and can be obtained for $9 per annum.

Ham Radio

Ham Radio is another excellent amateur radio publication, which has been published since the late 1960s. Its primary thrust is technical, and this is evident in the list of articles found (for example) in the March 1976 issue:

Crystal Controlled Oscillators
DT-500 RTTY Demodulator
WWVB Signal Processor
High Speed Divide by N Counters
Off the Air Transmitter Tuneup
VHF/UHF Receivers – How to Improve Them
5/8 Wavelength Vertical Antenna for Two Meter FM
Microprocessors – 8080 Output Instructions
High Performance Bench Power Supply

The non-technical editorial content is a good approximation to 0.0%; the quality of the technical articles is on a par with many engineering publications of the professional world. As is the case with *QST*, *Ham Radio* has many interesting and informative advertisements for amateur radio products.

You can subscribe for $10 per annum by writing to Ham Radio, Greenville NH 03048.

CQ

Another amateur radio publication is *CQ*, named after the greeting code used by amateur radio operators when looking for a contact. It is also a monthly publication, and its February 1976 issue was filled with technical and general interest information for amateur radio operators. On the technical side, this issue had an extended commentary on slow scan TV methods, short columns on technical pointers, an article on hardware modification of a commercial ham transceiver, etc. *CQ* also carries many advertisements for amateur radio products.

You can subscribe to *CQ* for one year @ $7.50. Write CQ, 14 Vanderventer Av, Port Washington NY 11050.■

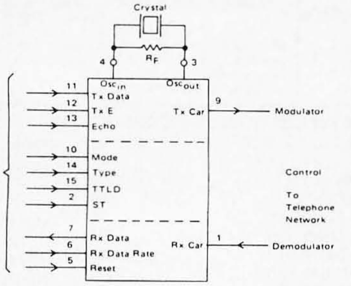
MC14412 UNIVERSAL MODEM CHIP

MC14412 contains a complete FSK modulator and de-modulator compatible with foreign and USA communications. (0-600 BPS)

FEATURES:
- .On chip crystal oscillator
- .Echo suppressor disable tone generator
- .Originate and answer modes
- .Simplex, half-duplex, and full duplex operation
- .On chip sine wave
- .Modem self test mode
- .Selectable data rates: 0-200, 0-300, 0-600
- .Single supply
  VDD=4.75 to 15VDC - FL suffix
  VDD=4.75 to 6 VDC - VL suffix

TYPICAL APPLICATIONS:
- .Stand alone - low speed modems
- .Built - in low speed modems
- .Remote terminals, accoustic couplers

MC14412FL................................. $28.99
MC14412VL................................. $21.74
6 pages of data........................ .60
Crystal for the above......$4.95



MC14411 BIT RATE GENERATOR.

Single chip for generating selectable frequencies for equipment in data communications such as TTY, printers, CRT s or microprocessors. Generates 14 different standard bit rates which are multiplied under external control to 1X, 8X, 16X or 64X initial value. Operates from single +5 volt supply. MC14411........................ $11.98
4 pages of data................................. .40
Crystal for the above.......................... $4.95

4X4X2 CROSSPOINT SWITCH. MC3416L is a pair of 4X4 Matrices of isulated SCR's triggered by a common selection matrix. Use for low loss analog switching such as telephone, data selection, etc.
MC 3416L..................................$11.95
9 Paaes of specs for 3416 ....$1.00

4 DIGIT COUNTER. MM74C926 is a 4 digit counter with 7 segment output. Carry output for cascading and intermal display select allows outputting of counter or set of internal latches. 3 to 6V operation. Great for clocks, event and frequency counters.
MM74C926 - with spec sheet....................$12.00

FOUR QUADRANT MULTIPLIER. MC1495L provides output as a linear product of two analog input. Use for frequency doubler, balanced modutar/demodulator, electronics gains control.
MC1495 L....................................$5.50
6 pages of specs..................................60
9 pages of applications............................90

TELETYPE CODE CONVERSION CHIP
MM5220BL converts 5 level Baudot into 8 level ASCII. Use this chip to make your old TTY talk to your new computer.
MM5220BL.................................. $18.00
Specs for the above...................... .30

MOS TIME BASE KIT.
Only 1" X 1.5". Input 5 to 15 VDC, output is 60HZ square wave for portable or mobile clocks. PC board is drilled! MTBK-60HZ...................... $5.88



78H05 Voltage regulator. Fairchild 5V, 5A, TO-3 regulator. Take care of those heavy current requirments without separate regulator/pass transistor combinations. Use it with the same ease of instalation as the 309K(same pin arrangement.)..........with specs................ $11.25



LM317 Voltage Regulator. 1.5A, 3 terminal adjustable regulator in TO-3 case. Adjusts from +1.2V to +37V. Complete overload protection. .1% load regulation, .01%/V line regulation. No need to stock assorted regulators - just stock resistors.....................$4.99
Specs for the above.................. .70

DATA BOOKS BY NATIONAL SEMICONDUCTOR
DIGITAL. Covers TTL, DTL, Tri-State, etc. $3.95
LINEAR. Covers amplifiers, pre-amps, op-amps, .. $3.95
LINEAR APPLICATIONS. Dozens of application notes and technical briefs covering the use of op-amps, regulators, phase locked loops and audio amps................ $3.25
CMOS Gates, Flip Flops, registers, functional blocks $3
VOLTAGE REGULATORS. A must for anyone making a power supply. Complete theory including transformers, filters, heat sinks, regulators, etc................. $3.00
MEMORY. Information on MOS and Bipolar memories: RAMS, ROMS, PROMS and decoders/encoders... ... $3.95
INTERFACE. Covers peripheral drivers, level translators, line driver/receivers, memory and clock drivers, sense amps display driver and opto-couplers.................. $3.95
(Outside U.S., add postage for 1.5lbs)

DATA BOOKS FROM FAIRCHILD.
uA Linear. 776 pages of data and applications for Fairchild linear ICs. Great value..................$4.25
MOS/CMOS/nMOS/pMOS/CCD. Data and applications on MOS and charge coupled devices including preliminary data on new and future offerings. Want to know about 16K charge coupled line addressable memories?.........$3.95

AMP L'ANNY Says

This Year, Tri-Tek is Celebrating too!
It's our 5th Anniversary of supplying you, our good customers, with quality parts and products.

RCA

GOLD CHIP
Linear Integrated Circuits

Brand new process by RCA in which the aluminum metalization has been replaced by gold. The chip is then hermetically sealed. What this means to you is unprecedented reliability and uniformity. Plastic parts that meet mil specs!!
Tri-Tek is proud to be the first to bring this new level of performance to you at SURPLUS PRICES. Why buy regrades??

CA301A..Improved,general purpose op-amp,8 pin dip..59¢
CA307...Super 741 op-amp. 8 pin dip...............52¢
CA324...Compensated quad op-amp, 14 pin dip.....$1.80
CA339A.. Low offset quad comparator. 14 pin dip...$1.59
CA741C..Famous general purpose op-amp, 8 pin dip.. 45¢
CA747C..General purpose dual op-amp, 14 pin dip... 82¢
CA748C..Externally compensated 741, 8 pin dip.......49¢
CA1458..General purpose dual op-amp. 8 pin dip..... 69¢
CA3401..Quad single supply (5-18V) op amp. 14 pin..89¢
-- -- -- -- -- -- -- -- -- -- -- --
Another super buy from RCA. CA555 timer. 8 pin dip. 59¢

INTEL Data Catalog. Contains latest information on all the famous INTEL micro-processor and memories............$4.00

IMPROVED Performance version of the famous 8080, 8 bit micro processor. 8080A.............................$34.95

SN75491....digit driver............................ 59¢
SN75492....digit driver............................ 59¢

UART AY5-1013A An ideal device for driving computer peripherals such as teletypes & video terminals. ...$9.45

DC to DC CONVERTER MODULE. Tiny potted module is a complete regulated output up-converter. 4.5V D. C. input provides approximately 14V @12 mA. Will operate down to 2.5V with reduced output. Size only 1" X ½" X ¼". With connection diagram.........................$1.25

NSL4944 LED. Current regulated, universal diffused-lens red LED lamp. A GaAsP solid-state high intensity LED encapsulated in a plastic package containing a current regulating IC that provides constant intensity over a wide voltage range. 2 to 18V, AC or DC. Use for indicator lamps, optical coupling, battery charging circuits, logic probes, almost any place you need a lamp. Long life, wide angle. No series resistor needed. Typical 13mA forward current.
NSL4944.......with panel mounting clip........... 89¢

D-A CONVERTER BY ZELTEX
8 bit precision hybrid circuit for use in controllers, timers, volt meters, etc. Molded plastic package with P.C. pins. Super buy on this better than usual subsystem. ZELTEX model ZD430. DAC-430.....................$4.95

NEW BOOK!!! "An Introduction to Microcomputers" This is the book which Fairchild Semiconductor Company called ".....the best darned introduction to the industry to date." Covers everything from basic concepts to a review of real microcomputers. IMC-001.............$8.00

TRI-TEK, INC.
6522 NORTH 43RD AVENUE,
GLENDALE, ARIZONA 85301
PHONE 602 - 931-6949

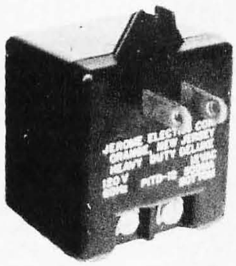
We pay shipping on all orders over $10 US, $15 foreign in US funds. Orders under $10, please add $1 handling. Please add insurance. Master Charge and Bank America cards welcome, ($20 minimum) Telephone orders may be placed 11AM to 5PM daily, Mon thru Fri. Call 602-931-4528. Check reader service card or send stamp for our latest flyers packed with new and surplus electronic components.

Continued from page 4

at a later date permitting later versions of the same board to have 32 K bytes of memory. Further, the firm in question stated that it would introduce a new twist to the marketplace by selling this product exclusively through retail computer stores in much the same way as high fidelity equipment is sold. The selection of power supply, keyboard, cassette recorder and TV monitor vendors is left to the purchaser as aided by the personal consultation of the retailer, just as selection of components of a high fidelity stereo system is often aided by an interactive session with a retailer.

- *Item:* As if to confirm the trend heralded by the phone call from "brand A," "brand B" called one week later to mention its version of the completely packaged and ready to go computer. The "brand B" product is similar to "brand A" in its confirmation of the trend to "no hassle" computing, but the nature of the system is a bit different. The "brand B" computer is a modular product with a traditional backplane design, 6800 processor and packaging assembly with power supply. The "brand B" computer looks inherently like a traditional minicomputer system available at very low prices which include (so it is said) a BASIC package. This "brand B" computer also comes in one form: assembled and completely burned in component cards which are mixed and matched to form a completed system. Again note the emphasis on "no hassle" hardware in the form of assembled and tested boards, with systems software bundled into the price, also under $1000 for a minimal system (8 K bytes) exclusive of RS-232 or 20 mA current loop IO terminal.

Both the "brand A" and "brand B" computers are not yet widely marketed, but the trend is clear: the products in the personal computing market place are becoming more refined and oriented to user satisfaction, yet still very reasonable in price. The "little IBMs" of the micro computer world will become the corporate giants of tomorrow by perceiving this trend and wrapping up a package of "customer satisfaction" and convenience.■



POWER SUPPLY MODULE

New, plug-in module. Plugs into AC outlet provides 12 volts AC at ½ amp by two screw terminals. Great for various clocks, chargers, adding machines, etc. New

$2.50 ea. 5/$11.00

POWER SUPPLY

LAMBDA 5VDC 74 AMP

LV-EE-5-OV $125.00

NJE 5/OUP-D5

5 VDC 32 AMP $75.00

CLOCK KIT $14.00

Includes all parts with MM5316 chip, etched & drilled PC board, transformer, everything except case.

SP-284 $14.00 each 2/$25.00



ASCII KEYBOARD, brand new w/TI ASCII chip inplace & data $45.00

COMPUTER GRADE LOGIC SUPPLY CAPS, BRAND NEW

47,000 Uf	25V	$2.00	ST	1,000	50	.90	AL
32,000	25	1.75	ST	3,300	35	1.25	AL
160,000	10	2.00	ST	1,600	20	.60	AL
66,000	10	2.00	ST	8,000	16	1.25	AL
1,000	60	.90	AL	500	6	.35	AL
2,000	55	1.00	AL	"ST" screw top . . . . "AL" axial			

5 VOLT 1 AMP REGULATED power supply kit for logic work. All parts including LM-309K $7.50

DUMMY LOAD RESISTOR, non inductive, 50 ohm 5 watt $1.00

"AA" NICAD CELLS brand new, fine biz for handy talkies $1.25 ea. 9/$9.00

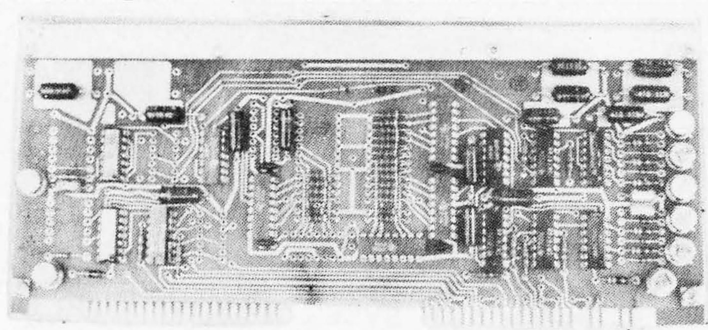
LINEAR by RCA, brand new, gold bond process

301	$ .60	747	$ .82	MM5314	$3.00
307	.52	748	.50	MM5316	3.00
324	1.80	1458	.96	7001	8.00
339A	1.60	3401	.80		
741	.50	555 timer	.60		

MEMORY SYSTEM $125.00

New memory system by Honeywell, small . . . measures only 9x4x1 inches. 1024 core memory, 1024 words with 8,9,10 bits/word. Random access, with all logic, register, timing, control, core select and sense functions in one package. New, booklet of schematics and data. Looks like a good beginning for a mini-computer. Limited supply on hand.

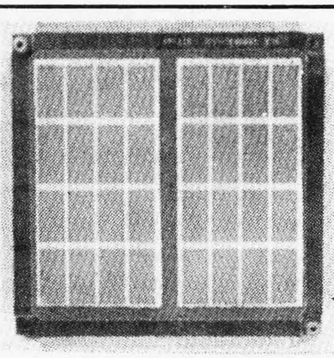
Ship wgt 3 lbs. #SP-79 . . . . $125.00



CORE MEMORY

Another brand new memory, ultra small. Measures only 4 x 4 inches with format on one plane of 32 x 32 x 16 (16,384). Only about 35 units of this on hand.

#SP-81 . . . . $20.00



FREE CATALOG SP-8 NOW READY

Please add shipping cost on above.

Meshna

PHONE 617–595-2275

MESHNA PO Bx 62 E. Lynn Mass. 01904

Continued from page 102

culator by either actually doing arithmetic (what?) or using an earlier version of the mechanical hand held adding machine. However, Ed forgot about the difference between base 16 and base 10, and the fact that readers without an exposure to traditional magazines for data processing (ie: large scale computers) probably had not heard of the "Hexadat." Since no one has come out with a way to manufacture and market several tens of millions of base 16 four function calculators at $8.95, we hexadecimal lovers are prevented from economically using a hand held base 16 electronic calculator. But in lieu of electronics, the Radix Precision "Hexadat" will serve excellently at times when you can't rely on your computer for accurate calculation.

You can acquire a Hexadat for $35.95 complete with the leather case and instruction manual. In case you are prejudiced against hexadecimal, the same company can help you out with the "Octadat" octal calculator at $14.95. Either way, contact Radix Precision at PO Box 13861, Atlanta GA 30324.■

April BOMB Results

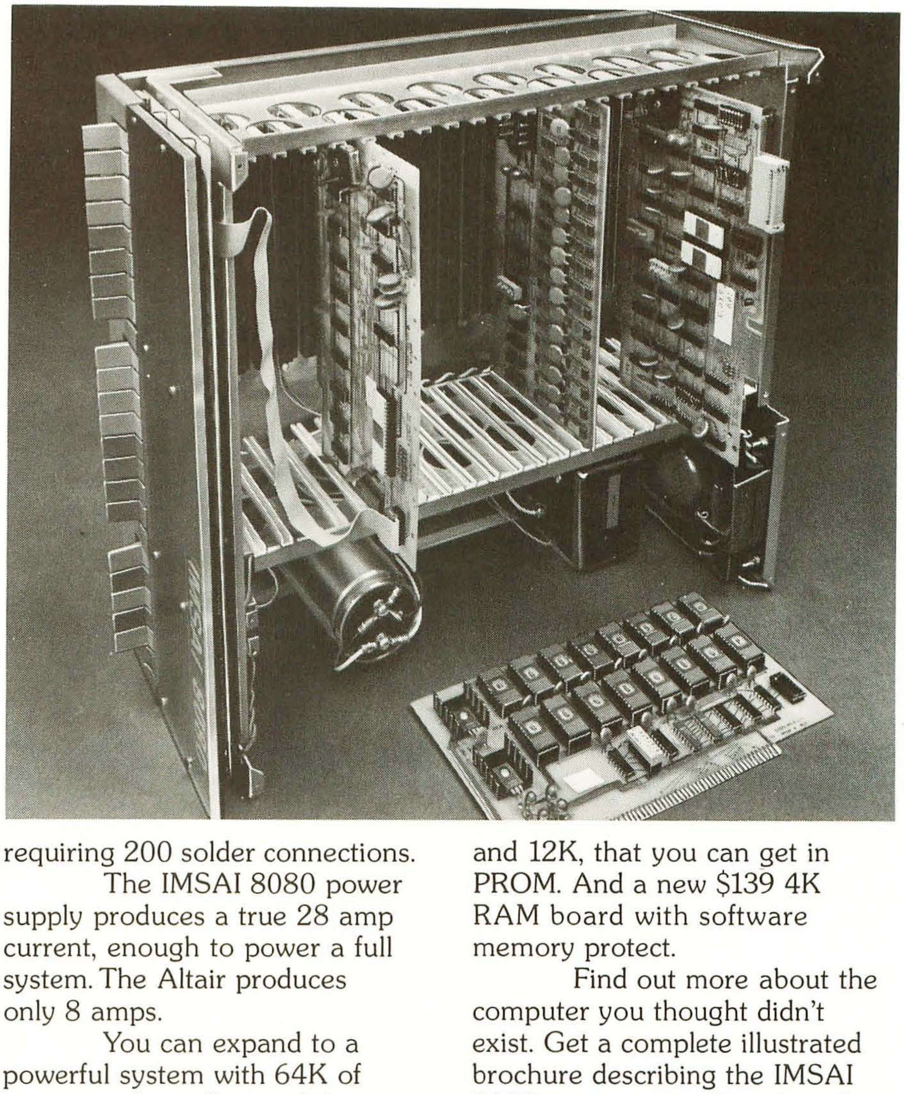
Winner of the $50 for most appreciated article in the April 1976 issue of BYTE is Theodor Nelson's "The Magic of Computer Languages." Runners up in the voting were Robert Wier and James Brown, who wrote "Design an On Line Debugger," and Don Lancaster's "How To Build a Memory With One Layer Printed Circuits."■

BOMB: BYTE's Ongoing Monitor Box

BYTE would like to know how readers evaluate the efforts of the authors whose blood, sweat, twisted typewriter keys, smoking ICs and esoteric software abstractions are reflected in these pages. BYTE will pay a $50 bonus to the author who receives the most points in this survey each month.

PAGE NUMBER	ARTICLE	LIKED: LEAST … BEST
6	Jones: Coincident Current Ferrite Core Memories	0 1 2 3 4 5 6 7 8 9 10
18	Anderson: Assembling a Sphere	0 1 2 3 4 5 6 7 8 9 10
22	Howerton: Explore an 8080 with Educator-8080	0 1 2 3 4 5 6 7 8 9 10
30	Wadsworth: Machine Language Programming for the "8008"—Chapter 1	0 1 2 3 4 5 6 7 8 9 10
40	Baker: Put the "Do Everything" Chip into Your Next Design	0 1 2 3 4 5 6 7 8 9 10
46	Suding: Why Wait?	0 1 2 3 4 5 6 7 8 9 10
54	Hayes: Surplus Electronics in Tokyo and Manila	0 1 2 3 4 5 6 7 8 9 10
58	Hogenson: Make Your Own Printed Circuits	0 1 2 3 4 5 6 7 8 9 10
64	Lerseth: A Plot is Incomplete Without Characters	0 1 2 3 4 5 6 7 8 9 10

If you thought a rugged, professional yet affordable computer didn't exist,



think IMSAI 8080.

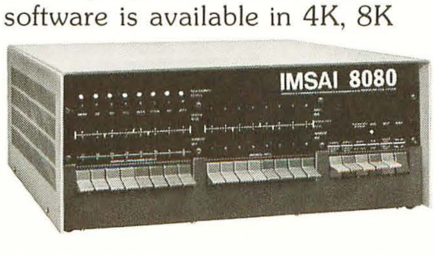
Sure there are other commercial, high-quality computers that can perform like the 8080. But their prices are 5 times as high. There is a rugged, reliable, industrial computer, with high commercial-type performance. And prices that are competitive with Altair's hobbyist kit. The IMSAI 8080. Fully assembled, it's $931. Unassembled, it's $599. And ours is available now.

In our case, you can tell a computer by its cabinet. The IMSAI 8080 is made for commercial users. And it looks it. Inside and out! The cabinet is attractive, heavy-gauge aluminum. The heavy-duty lucite front panel has an extra 8 program controlled LED's. It plugs directly into the Mother Board without a wire harness. And rugged commercial grade paddle switches that are backed up by reliable debouncing circuits. But higher aesthetics on the outside is only the beginning. The guts of the IMSAI 8080 is where its true beauty lies.

The 8080 is optionally expandable to a substantial system with 22 card slots in a single printed circuit board. And the durable card cage is made of commercial-grade anodized aluminum. The Altair kit only provides 16 slots maximum in four separate sections, each section requiring 200 solder connections.

The IMSAI 8080 power supply produces a true 28 amp current, enough to power a full system. The Altair produces only 8 amps.

You can expand to a powerful system with 64K of memory, plus a floppy disk controller, with its own on board 8080 – and a DOS. An audio tape cassette input device, a printer, plus a video terminal and a teleprinter. These peripherals will function with an 8-level priority interrupt system. IMSAI BASIC software is available in 4K, 8K and 12K, that you can get in PROM. And a new $139 4K RAM board with software memory protect.



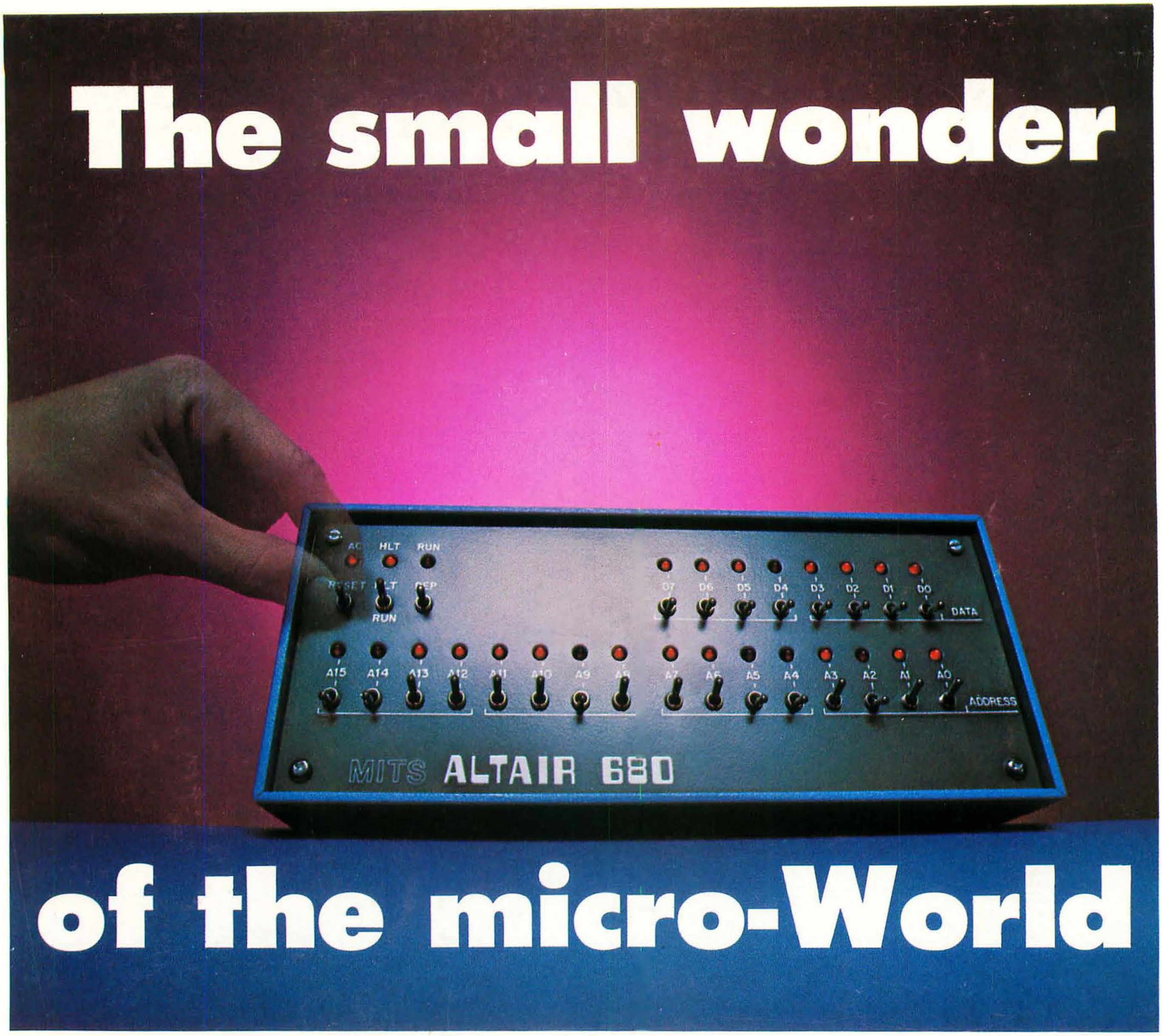
Find out more about the computer you thought didn't exist. Get a complete illustrated brochure describing the IMSAI 8080, options, peripherals, software, prices and specifications. Send one dollar to cover handling to IMS. The IMSAI 8080. From the same technology that developed the HYPERCUBE Computer architecture and Intelligent Disk systems.

Dealer inquiries invited.

IMS

IMS Associates, Inc.
1922 Republic Avenue
San Leandro, CA 94577
(415) 483-2093

Dept. B-7



Measuring just 11″ wide x 11″ deep x 5″ high, and weighing a mere 7 pounds, the Altair™ 680b is a complete, general-purpose computer.

The secret to this revolutionary, small computer is its CPU board. This double-sided board fits along the bottom of the Altair case and plugs directly into the front panel board. It contains the new 6800 microprocessor, 1,024 bytes of RAM memory, a 256 byte PROM monitor, provisions for 768 bytes of additional PROM or ROM, and a single Interface port with a Motorola ACIA serial interface adapter which can be configured either RS-232 or TTY. A five level Baudot interface option is also available.

The Altair 680b can be programmed from front panel switches, or it can be interfaced to a video display terminal, or teletypewriter. Three additional circuit boards can be plugged inside the Altair 680b for further memory and interface expansion. The first of these boards now under development is an 8K RAM memory board.

Software already developed includes a resident two pass assembler and 8K BASIC. The Altair 680b is also compatible with Motorola 6800 software.

The Altair 680b is ideal for hobbyists who want a powerful computer system at an economic price. Altair 680b owners qualify for membership in the Altair Users Group, and like other Altair owners, they receive a complimentary subscription to **Computer Notes** and complete factory support.

NOTE: Altair is a trademark of MITS, Inc.

PRICES:

Altair 680b kit with complete, easy-to-understand assembly manual, operator's manual, and programming manual $466
Assembled Altair 680b $625
Baudot option .... $ 42
8K RAM memory board and software prices to be announced soon.

MAIL THIS COUPON TODAY

☐ Enclosed is a check for $________
☐ BankAmericard #________ or Master Charge #________
☐ Altair 680b ☐ Kit ☐ Assembled
enclose $8 for postage and handling
☐ Please send free information package.
NAME________
ADDRESS________
CITY________ STATE & ZIP________

Price, specifications subject to change. Please allow up to 60 days for delivery.

mits

2450 Alamo SE/Albuquerque, NM 87106, 505-243-7821